Stellar Classification Using PCA and Machine Learning on the SDSS17 Dataset

Jashwanth Reddy Earla
Student ID: 40271577

GitHub Link: INSE-6220/stellar_classification_PCA.ipynb at main · JashwanthReddyE/INSE-6220 (github.com)

Abstract---This project delves into the classification of celestial objects, including stars, galaxies, and quasars, using the "Stellar Classification Dataset - SDSS17." Employing Principal Component Analysis (PCA), we aim to simplify the intricate spectral data and identify key components crucial for efficient classification. By integrating machine learning techniques, our approach seeks accurate categorization of these celestial entities based on their spectral characteristics. The project not only enhances the interpretability of the dataset through feature extraction but also contributes to a broader understanding of the universe. Through the synergy of PCA and machine learning, we aspire to unravel complex patterns within the SDSS17 dataset, offering insights that can inform future astronomical observations and enrich our comprehension of the cosmos.

Keywords—***Principal Component Analysis (PCA), Machine Learning, Stellar classification dataset, classification.***

1. INTRODUCTION

Stellar classification, a fundamental pursuit in astronomy, categorizes celestial objects based on their spectral characteristics, offering insights into the diverse composition of the universe. The "Stellar Classification Dataset - SDSS17" serves as a crucial resource, providing spectral data for stars, galaxies, and quasars. In this project, Principal Component Analysis (PCA) is employed to streamline the complexity of the dataset, enhancing the efficiency of classifying these cosmic entities.

Stars, galaxies, and quasars, each with their unique cosmic signatures, unveil the universe's rich tapestry. The historical cataloging of stars, coupled with advancements in telescopic technology, has expanded our understanding beyond our own galaxy to countless others. The revelation of separate galaxies, exemplified by Andromeda, marks a transformative era in astronomical exploration.

At the intersection of historical insights and modern analytical techniques lies our research, aiming to refine the classification of celestial entities. By leveraging the "Stellar Classification Dataset - SDSS17" and employing PCA, we strive to contribute to a deeper understanding of the cosmos, illuminating the mysteries that have captivated humanity for centuries.

2. PRINCIPAL COMPONENT ANALYSIS(PCA)

Data sets are often huge and complex in the real world. These characteristics make data presentation, data processing, and storage excessively expensive. On the other hand, the manifold hypothesis states that low-dimensional manifolds can be used to find actual high-dimensional data. Therefore, techniques for reducing real-world data sets to lowdimensional spaces are used to simplify interpretation.
A multivariate technique called principal component analysis is used to simplify a large number of related variables into a feasible number of primary components, which are independent variables that attempt to capture as much variability in the original variables as possible. It is a statistical tool for feature extraction. The method keeps the diversity of the data by creating new variables that are linear combinations of the original parameters. Principal Components (PCs), the new variables, can be compared to additional data coordinates. The first PC captures the largest variance in the data by shortening the distance between the data and its projection onto the PC. The subsequent PCs, which are independent of or uncorrelated to the earlier ones, minimize this distance as well.

A. Steps for solving PCA:

A data matrix is a set of data with n rows and p columns that is used in PCA.
1) Centre data: Initially, we need to find out the average for each column. Now, the average is subtracted from each piece of data entered in that column.
Y = HX ❼ (1) is the centering matrix with nxp dimensions.
2) Covariance matrix: The covariance matrix is constructed from the centered matrix(Y) using the formula S = Y T 1/n Y ❼ (2)
The S matrix has pxp dimensions.
3) Eigen decomposition: Determine the eigenvectors and eigenvalues using eigen decomposition.
Here, A* is S = A A T ❼ (3)
4) Principal component: Compute the transformations using the n x p matrix (Z). The columns of Z represent the PC, whereas the rows of Z represent the observations. The

number of PCs is equivalent to the dimension of the algorithm in the initial data matrix.
Z = (Z1', Z2', Z3',.....ZI',.....ZP') = Y A ❼ (4)

III. Classification Algorithms

Classification is a technique for structuring our data into a reasonable number of different classes, each with its own labels.
Unsupervised learning and supervised learning are the two methods that machine learning classifiers can be used in. In supervised learning, we use labelled data to train the computer, whereas unsupervised learning is a machine learning technique that doesn't need supervision. Two different classes of supervised machine learning techniques are regression and classification. We have applied the labelled data in this project. We will however employ the logistic regression, k nearest neighbors (lazy learner classifier), and random forest, decision tree classification techniques on this data set also.

Logistic Regression:

Logistic regression is a technique for determining the likelihood of a specific result depending on an input variable. The most popular kind of logistic regression models a binary outcome, such as true or false.
When there are more than two possible outcomes, you can predict what will happen by using multinomial logistic regression. Finding out whether a fresh sample fits into a group the best can be done using logistic regression.

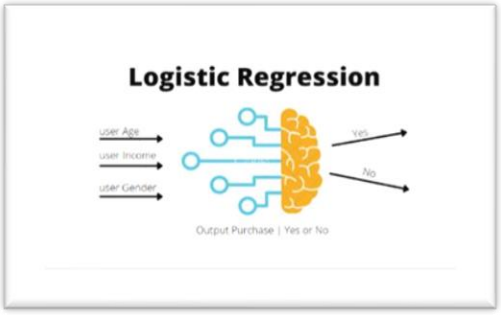


Figure 1[3]

Decision Tree:

Decision tree uses a tree structure to develop classification or regression models. It uses an extensive and mutually exclusive set of if-then rules for classification. The training data is used to learn the rules one at a time. Every time a rule is learned, the tuples it applies to are removed. This approach is continued on the training set until a termination requirement is met. The benefits of a decision tree are that it requires very little data preparation and is easy to comprehend and visualise.
Decision Tree functions well on training data, but an overfitted model performs terribly on not observed data due to over-fitting of the data, which leads to an excessive number of branches that may reflect irregularities caused by noise or outliers. Pre-pruning, which halts tree growth before it starts, or post-pruning, which prunes branches off of already-grown trees, can prevent this.

KNN:

The k-Nearest Neighbour algorithm is a lazy learning technique that stores all instances of training data points in ndimensional space. When a discrete data set with unknown properties is received, the algorithm analyses the closest k preserved instances (nearest neighbours) and returns the most frequent class as the prediction; for real-valued data, the algorithm gives the mean of the K-nearest neighbours. The idea of similarity is encapsulated by KNN (sometimes called distance or proximity).

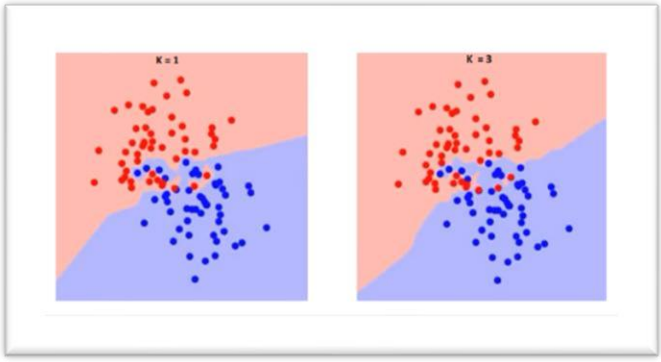


Figure 2[4]

From the figure 2, it can be observed that the boundary becomes smoother with increasing value of K. If K increases to infinity, it finally becomes all red or blue depending on the total majority.

Random Forest:

A vast number of decision trees are constructed during training to create random forests, often referred to as random choice forests, which are an ensemble learning strategy for classification, regression, and other problems. The class that the majority of trees select is the output of the random forest for classification problems. When performing a regression task, the mean or average forecast of each individual tree is returned. The issue of decision trees overfitting their training set is tackled by random decision forests.
Although they frequently outperform decision trees, random forests are less accurate than gradient-enhanced trees. Data features, on the other hand, may have an effect on their performance.

ADA Boost:

It is a supervised machine learning algorithm used in both classification and regression. Any machine learning algorithm's performance can be improved with AdaBoost. It works well with weak learners. These are models whose classification accuracy is slightly better than random chance. Decision trees with one level works optimally with AdaBoost and is used most extensively.

IV. Dataset Description

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 datasat aims to classificate stars, galaxies, and quasars based on their spectral characteristics.

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(from which some are removed for a good fit of PCA) and 1 class column which identifies it to be either a star, galaxy or quasar.

This data considered in our study has 6 features to predict the stellar. The following pie chart shows the classification data.

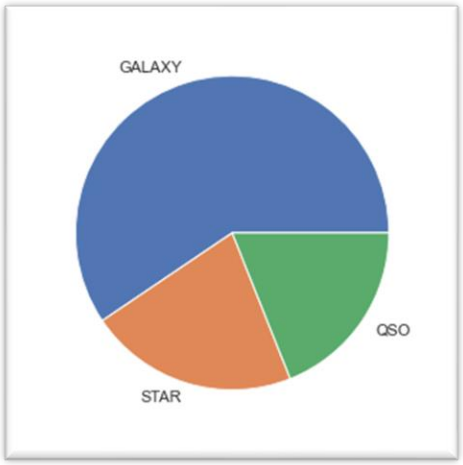
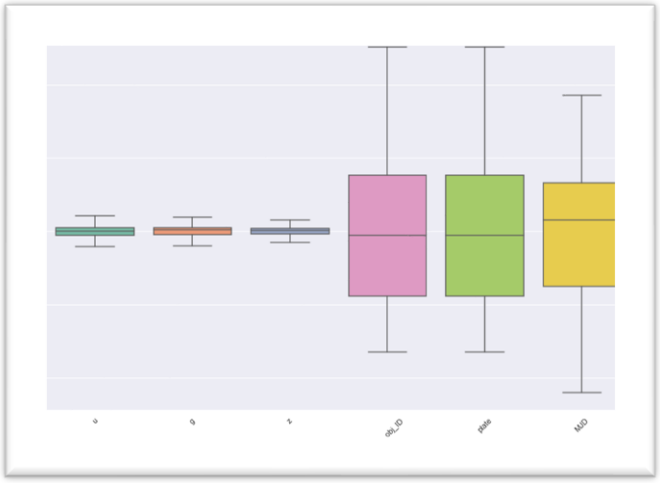


Figure 3

As in figure 4, it shows a box plot of the data. “spec_obj_ID”, “plate” and “u” attribute seem to be approximately normally distributed. And “MJD”, “g”, “z”



Here in Figure 8, the pairs plot shows the relationship between all the attributes individually and with each other. It is a visual tool to better understand the trends of a large dataset.

Figure 4

seems to be negatively skewed with outliers. Outliers denote the lines that extend from each box that show the remaining data, with dots outside the line borders.

Figure 5 represents all the observations in a strip plot, which is an addition to the box plot. It displays all the data points and its spread.

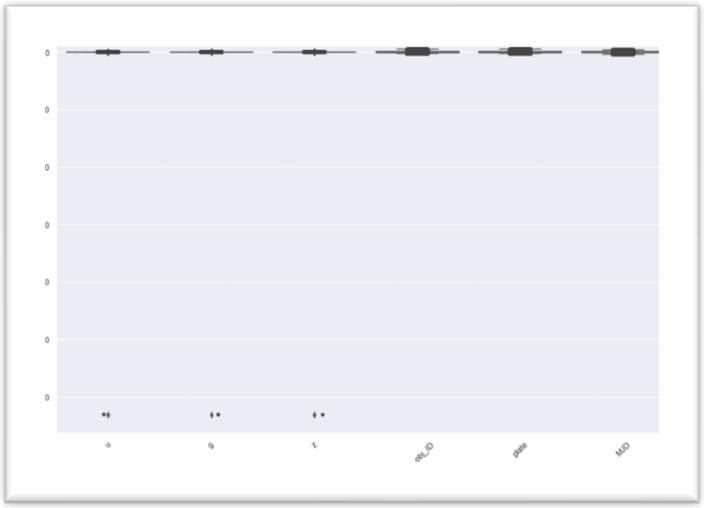


Figure 5

The covariance matrix is a square matrix that represents the degree of correlation between any two elements of a random vector. Figure 6, shows correlation between the columns. It is for a fact that ‘u’, ‘g’, ‘z’ are correlated, we can ‘spec_obj_id’, ‘plate’, ‘MJD’ have a strong correlation.

Figure 6

Here in Figure 7, the pairs plot shows the relationship between all the attributes individually and with each other. It is a visual tool to better understand the trends of a large dataset.

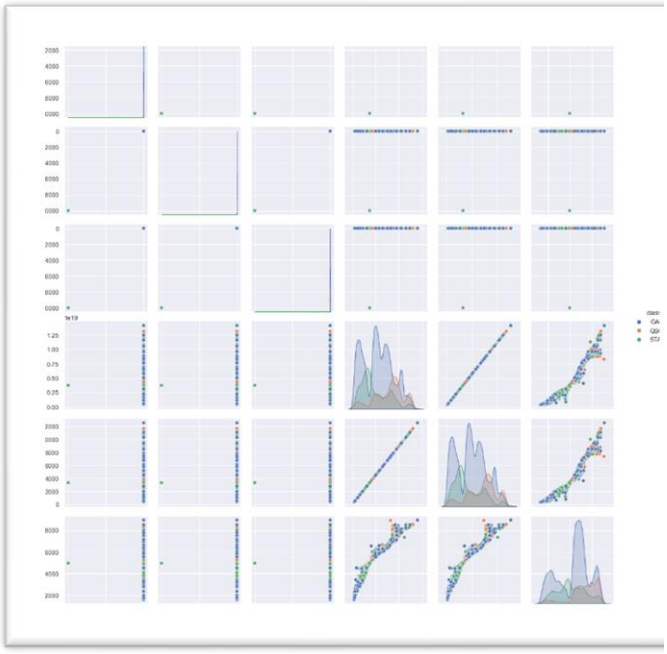


Figure 7

V. PCA RESULTS:

The dimensions of the data set were reduced using PCA. The data set was reduced using the methods in Section ii to r features with r<6 from six features. The eigenvector matrix is used to condense the n x p data set (A). Using PCA, the eigenvector matrix that was produced for the occupancy data was:

```
[[-4.39677466e-01,-3.74285747e-01,2.36
939327e-02,6.75158525e-01,-4.58468612e
-01,4.43553346e-06],
 [-4.43103060e-01,-3.70239431e-01,5.58
000258e-04,5.91275072e-02,8.14300969e-
01,1.32660895e-05],
 [-4.42199957e-01,-3.71013106e-01,-2.5
2520457e-02,-7.34479730e-01,-3.5596394
9e-01,-1.75985465e-05],
 [-3.72933276e-01, 4.43312170e-01, -4.
05144946e-01,1.56140981e-02,-2.2153989
7e-03,-7.07109579e-01],
 [-3.72933141e-01,4.43312157e-01,-4.05
155630e-01,1.55908162e-02,-2.23667332e
-03,7.07103983e-01],
 [-3.69654224e-01,4.38327355e-01,8.188
44985e-01,-2.67872309e-02,-4.68992483e
-04,5.77766517e-06]]
```

Each principal component's (PC) eigenvalues correspond to the variables that each PC extracts from the data. Visual representations of the variance that each PC accounts for include a Scree plot and a variance plot. The following equation is used to calculate the percentage of variation accounted for by the jth PC:

$$l_j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100\% \; for \; j = 1,2,.,p$$

The variance from the first two PCs are: l1 = 51.4% and l2 = 47.8%The first two PCs combined account for 99.96% of the total variance in the dataset. Fig. 8 and 9 explains the relationship between principal components and variance. The scree plot shows that the elbow curve is at the PC2.

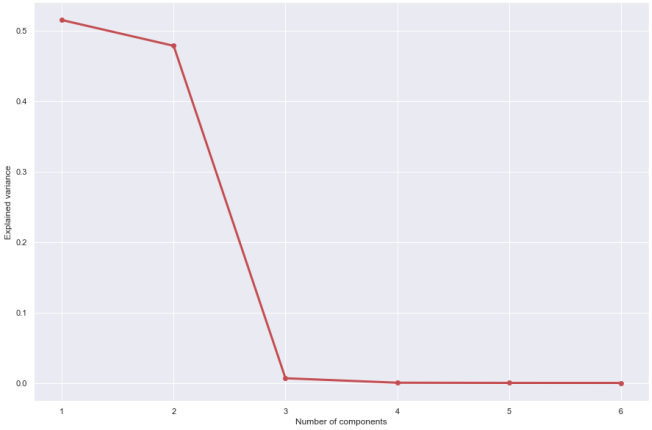


Figure 8

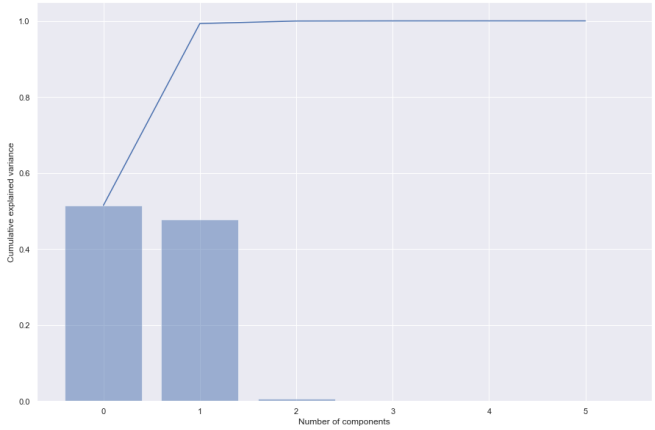


Figure 9

According to the scree plot and explained variance of the two first PCs, the dimension of the data can be reduced to r=2. The principal components 1: Z1 = -0.439677X1-0.443103X2-0.442200X3-0.372933X4-0.372933X5-0.369654X6

principal components 2: Z2 = -0.374286X1-0.370239X2-0.371013X3+0.443312X4+0.443312X5+0.438327X6

In PC1, X6 has less contribution in comparison to others. Whereas in PC2, X6 has the second highest contribution among the other values. In both PC1 and PC2, there are no negligible components that must be excluded from the equations.

In Figure 10 , the contribution of each variable to PC1 AND PC 2 are shown using a PC coefficient plot. It graphically shows the coefficients in Z1 and Z2.

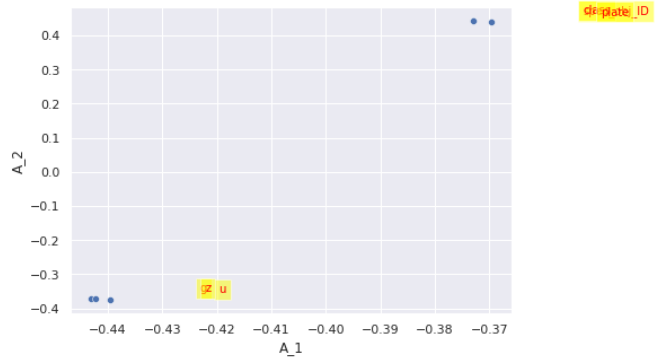


Figure 10

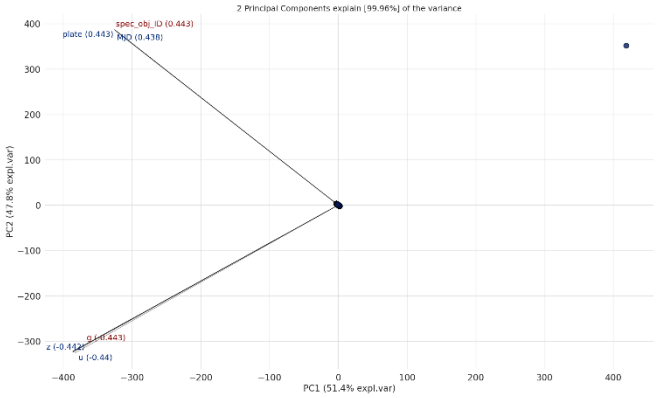


Figure 11

In figure 11, a better visual representation of the equations are plotted using a biplot. The x and y axis are PC1 and PC2 respectively. The observations are marked as colored dots and the rows of the eigen vector matrix are used as vectors in this diagram. The smaller the angle between the vectors and the axis, the larger is the contribution.

VI . CLASSIFICATION RESULTS:

Pycaret is used the classification and visual presentation of the results. The term "training data" describes information used to hone the algorithm. This will help the algorithm create labels. 70% of the data were used for training. The remaining information is used to evaluate the algorithm's precision and effectiveness. The aim is to lower stress levels. utilizing the built-in "compare model". It compared every model in the library and gave back the cross validated performance measures on average. The output of the best model using the built-in technique was "Random forest classifier."

Using stratified cross validation, the tune model() function adjusts a model's hyperparameters on a predefined search space. The result shows a score grid with fold-by-fold values for Accuracy, AUC, Recall, Precision, and F1.

Recall indicates the percentage of positives that are found for each class, whereas precision measures the percentage of forecasts that are just positive. Confusion matrices and area under the curves are used to visually picture the precision and recall outcomes (AUC). It draws attention to the areas where the models have the most difficulty making accurate predictions. The model performs better at predicting class 0 and 1(occupied or not occupied) when AUC score is higher.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.8719	0.9533	0.8719	0.8699	0.8701	0.7695	0.7704	5.2350
lightgbm	Light Gradient Boosting Machine	0.8714	0.9553	0.8714	0.8696	0.8694	0.7681	0.7693	1.4190
et	Extra Trees Classifier	0.8628	0.9498	0.8628	0.8606	0.8601	0.7519	0.7535	1.9610
gbc	Gradient Boosting Classifier	0.8402	0.9382	0.8402	0.8382	0.8341	0.7078	0.7123	15.0700
dt	Decision Tree Classifier	0.8099	0.8394	0.8099	0.8102	0.8100	0.6633	0.6634	0.3160
ada	Ada Boost Classifier	0.7821	0.8419	0.7821	0.7722	0.7663	0.5939	0.6033	1.3420
lda	Linear Discriminant Analysis	0.7519	0.8307	0.7519	0.7416	0.7372	0.5425	0.5498	0.1190
ridge	Ridge Classifier	0.7392	0.0000	0.7392	0.7357	0.7012	0.4989	0.5228	0.1000
knn	K Neighbors Classifier	0.7021	0.7747	0.7021	0.6945	0.6856	0.4242	0.4368	0.2130
nb	Naive Bayes	0.6022	0.5999	0.6022	0.4562	0.4812	0.0620	0.1194	0.0910
lr	Logistic Regression	0.5941	0.4462	0.5941	0.3530	0.4429	0.0000	0.0000	1.7270
dummy	Dummy Classifier	0.5941	0.5000	0.5941	0.3530	0.4429	0.0000	0.0000	0.1250
svm	SVM - Linear Kernel	0.3594	0.0000	0.3594	0.1660	0.2183	0.0000	0.0000	2.6710
qda	Quadratic Discriminant Analysis	0.2260	0.5112	0.2260	0.0771	0.1035	0.0197	0.0327	0.1340

Figure 12

Figure 12, shows comparison of models and figure 13 shows comparison of models after PCA. After PCA, “Light gradient boosting machine” algorithm is the best model.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.7118	0.7884	0.7118	0.7107	0.6867	0.4214	0.4489	0.9610
knn	K Neighbors Classifier	0.7003	0.7612	0.7003	0.6897	0.6841	0.4213	0.4332	0.2590
gbc	Gradient Boosting Classifier	0.6962	0.7641	0.6962	0.6938	0.6662	0.3852	0.4150	7.4060
rf	Random Forest Classifier	0.6864	0.7672	0.6864	0.6733	0.6745	0.4085	0.4143	3.3770
et	Extra Trees Classifier	0.6764	0.7594	0.6764	0.6634	0.6659	0.3946	0.3987	1.8170
ada	Ada Boost Classifier	0.6562	0.7137	0.6562	0.6526	0.6316	0.3307	0.3459	0.8690
qda	Quadratic Discriminant Analysis	0.6419	0.6677	0.6419	0.6451	0.5586	0.2106	0.2751	0.1470
lr	Logistic Regression	0.6171	0.6500	0.6171	0.5975	0.5278	0.1333	0.1906	1.5680
dt	Decision Tree Classifier	0.6086	0.6552	0.6086	0.6097	0.6091	0.3082	0.3082	0.2260
lda	Linear Discriminant Analysis	0.6050	0.5957	0.6050	0.4719	0.4882	0.0738	0.1308	0.1420
ridge	Ridge Classifier	0.5978	0.0000	0.5978	0.3788	0.4531	0.0171	0.0236	0.1480
svm	SVM - Linear Kernel	0.5974	0.0000	0.5974	0.5001	0.4545	0.0189	0.0699	0.2620
dummy	Dummy Classifier	0.5941	0.5000	0.5941	0.3530	0.4429	0.0000	0.0000	0.1460
nb	Naive Bayes	0.5874	0.5726	0.5874	0.4903	0.5138	0.1453	0.1673	0.1490

Figure 13

The confusion matrices were then created to see if the algorithms had trouble differentiating between classes. The model's prediction is done by the horizontal axis of the confusion matrix, while the true label is done by the vertical axis. The following figures 14,15 shows the confusion matrix for random forest classifier and lightgbm respectively.

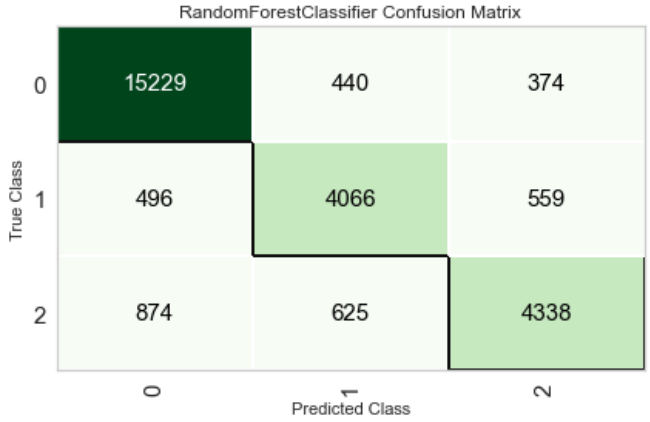


Figure 14

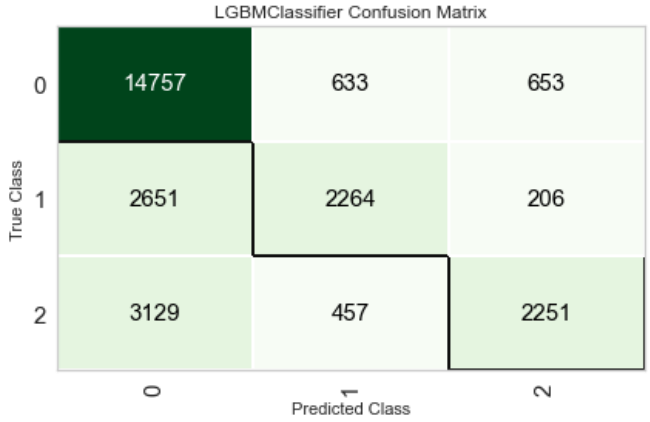


Figure 15

The figures 16,17 show the ROC curves plotted for each model. The area under the curve AUC gives a summary of the ROC curves. If the area under the curve is 0.5, they are random classifiers and if they are 1.0, they are perfect classifiers.

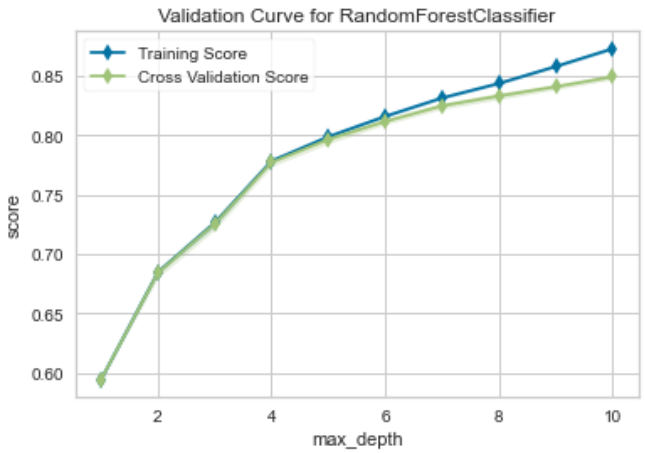


Figure 16

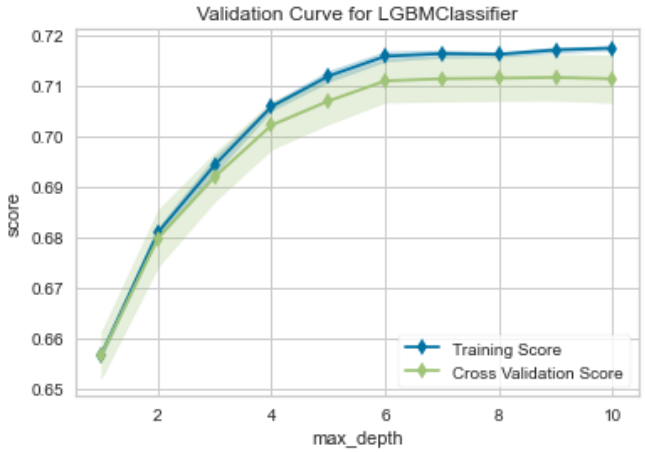


Figure 17

VII . EXPLAINABLE AI WITH SHAPLEY VALUES:

The Shapley Additive Explanations (SHAP) plot for the prediction of occupancy of the room is described above. The elements of the summary plots (y-axis) are arranged according to their mean absolute SHAP values. Each point's position on the x-axis represents the impact of that attribute on the classification result. The values of the attributes are represented by color for example in components 1,2 and 3.

VIII . CONCLUSION:

Finally, the Stellar Classification Dataset - SDSS17 was evaluated to the application of PCA and three widely used Machine learning classification techniques. Six variables in the data is described about the stellar. The features of the dataset were subjected to PCA in the first step. It was discovered that 99.96% of the variance in the data is captured by the first two PC1 and PC2. The dataset was thus split into two halves. Visual representations were created after a detailed examination into the factors that affect the first two PCs. Decision tree, Random Forest and Light gradient boosting Machine was applied to predict

the best model and accuracy with which the data could be classified. Random forest performed the best and for classification, with PCA model, Light gradient boosting machine algorithm gave the best results.

IX . REFERENCE:

1. Abdurro'uf et al., The Seventeenth data release of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar and APOGEE-2 DATA (Abdurro'uf et al. submitted to ApJS) [arXiv:2112.02026]
2. fedesoriano. (January 2022). Stellar Classification Dataset - SDSS17 Stellar Classification Dataset - SDSS17 (kaggle.com)
3. Develop a Logistic Regression Machine Learning Model | by Haq Nawaz | Dev Genius
4. KNN Algorithm | Latest Guide to K-Nearest Neighbors (analyticsvidhya.com)
5. Advanced Statistical Approaches to Quality, Concordia University.