Machine Learning Based Approaches for the Classification of Astronomical Objects Using Sloan Digital Sky Survey (SDSS)

Melakeneh Gedefaw, Natalie Kellner, David Macias, Kimberly Hanson

1. Introduction

Throughout time, humanity has tried to classify the objects seen in the night sky to bring order and understanding to our world. With the advent of large sky surveys such as SDSS (and others), the amount of data increases daily, exceeding our ability to process in traditional ways. Machine Learning (ML) offers a path to process and analyze this information and allows more detailed classifications to be performed, deepening our understanding of the Universe.

1.1 Motivation

The advancement in optical and computer technology contributed to the dramatic growth in the observational and computational aspects of astronomy and astrophysics over the last two decades (Rodriguez et al., 2022). This has resulted in an overwhelming prevalence of large datasets (i.e., terabytes — trillions of bytes — of data), that require a new method of analysis (Bailer Jones, 2023). Approaching the identification, labeling, and classification of astronomical objects manually is no longer feasible compared to the shared amount of data being produced. Thus, extremely fast, less biased, and reliable analysis mechanisms are needed to improve the understanding of our universe from the existing and future astronomical data that is being doubled each year. Hence, the team has been motivated to investigate, evaluate, and recommend a machine-learning approach for the identification, prediction, and classification of astronomical objects.

1.2 Problem statement

The prevalence of astronomical backlogged data is associated with the challenge of "big data," where immense volume, high dimensionality, disparate variable, and missing observations are its popularity. Despite various incredible efforts that helped to give solutions to the existing problem, a great deal of Machine Learning (ML) based efforts need to be employed to exploit the existing data and acquire comparable knowledge to the level of the efforts invested to innovate their collection for humanity. In this regard few initiatives or interests have been observed in comparing different ML classification and prediction algorithms to identify astronomical objects using SDSS data at the student's project scale. Thus, the project aims to test ML classification algorithms peculiar to astronomy and recommend them for future use.

1.3 Objective

The objective of this project is to apply and compare the performance of different machine learning algorithms to classify astronomical objects.

1.3.1. Specific Objectives

a. Compare the performance of machine learning algorithms before and after data dimensionality reduction
b. Select the best-performing classification algorithm for astronomical object classification and
c. Perform hyperparameter fine-tuning and accuracy test

2. Methodology

2.1 Data

Astronomical data were acquired from the Sloan Digital Sky Survey (SDSS). The data consists of 10,000 instances of five attributes and three classes of objects: Galaxy, Star and Quasar. The attributes used in the analysis are the magnitudes of the electromagnetic waves (bands) used to differentiate objects using different Machine Learning algorithms (e.g., Logistic regression, SVM, KNN.) The bands include infrared (e.g., Ultraviolet ray filtered at 3545 Ångstroms to the visible bands (e.g., green band) to near and far infrared (Table 1).

Feature	Description	Data type
Ultraviolet ray (u)	It is found within the wavelength range of (e.g., 0.30 µm - 0.38 µm (1µm = 10^6m) in the electromagnetic spectrum and filtered at 3543 Ångstroms (Å) to discriminate the astronomical objects	Float 64
Green (g)	It is found within the visible electromagnetic spectrum of 0.5 µm - 0.578 µm and filtered at 4770 Ångstroms (Å) to discriminate the astronomical objects.	Float 64
Red (r)	It is found within the visible electromagnetic spectrum of 0.62 µm - 0.7µm and filtered at 6231 Angstroms (Å) to discriminate the astronomical objects.	Float 64
Near-Infrared (i)	It is found within the infrared (IR) spectrum of the electromagnetic wave (0.7 µm -100 µm) with a bandwidth of 0.7 to 1.5 µm. To discriminate the objects, the NI was filtered with 7625 Ångstroms (Å).	Float 64
Infrared (z)	It is found within the infrared (IR) spectrum of the electromagnetic wave (0.7 µm -100 µm) with a bandwidth of 0.7 to 1.5 µm. To discriminate the objects, the z band was filtered with 9134 Ångstroms (Å).	Float 64
Class	Astronomical objects (I.e., Starts, Galaxy and QBR) identified with the specific thresholds of the bands	Object

Table 1. Description of features and classes

2.2 Data preparation

Data cleansing was performed on the dataset where any probabilistic records and classes that exhibited insufficient or unbalanced data were removed. In relation to this, object classes whose records were encoded as text were converted to integers. Data transformations including feature scaling (e.g., standardization and normalization) were implemented on the datasets. Using untransformed and transformed data, basic statistics of five number summary, correlation matrix, and distribution graphs were prepared (Figures 1 & 2).

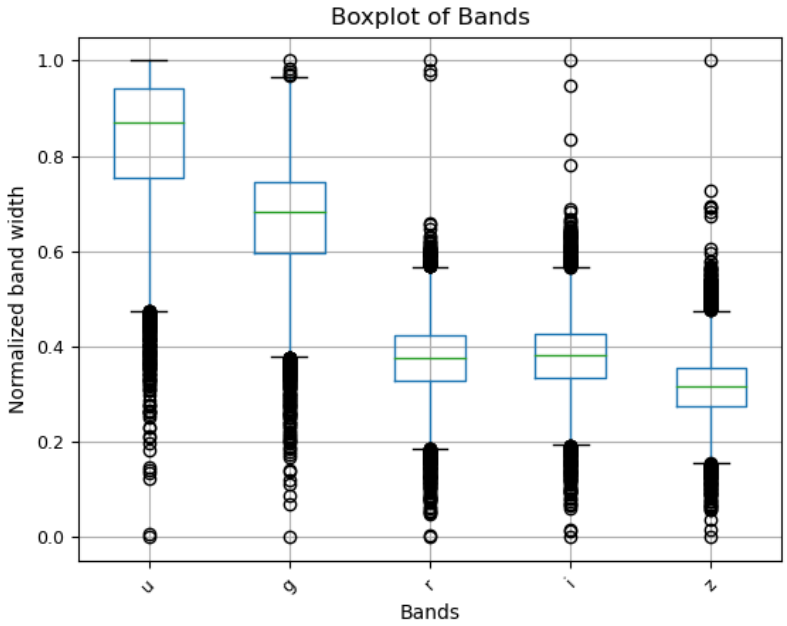


Figure 1. A five number summary of the dataset features (after normalization)

As can be seen from figure 1, the features exhibited outlier records, but differed in mean values, except red and near infrared bands where they showed nearly equal mean values of normalized wavelength. This might be due to the proximity of their wavelengths in the electromagnetic spectrum. This relationship is easily seen in the correlation heatmap where their correlation is strong and positive. (Figure 2)

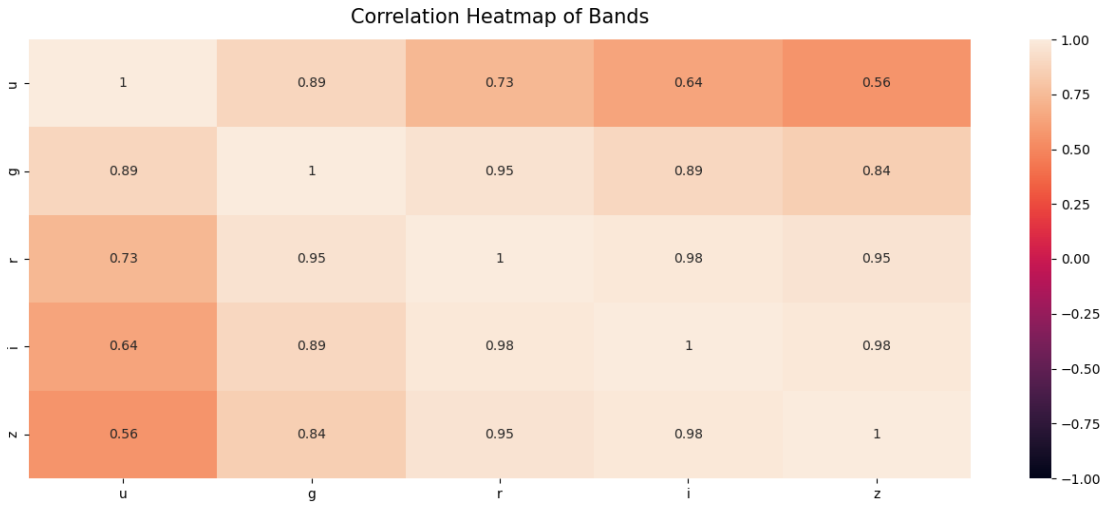


Figure 2. Correlation Heat map of bands.

A positive and strong correlation was also observed between green and red bands, which might also be related to the nearness of each band within the visible range of the magnetic spectrum (Annex I). Figure 3 compares the magnitudes of the infrared and ultraviolet wavelengths between the 3 types of objects.

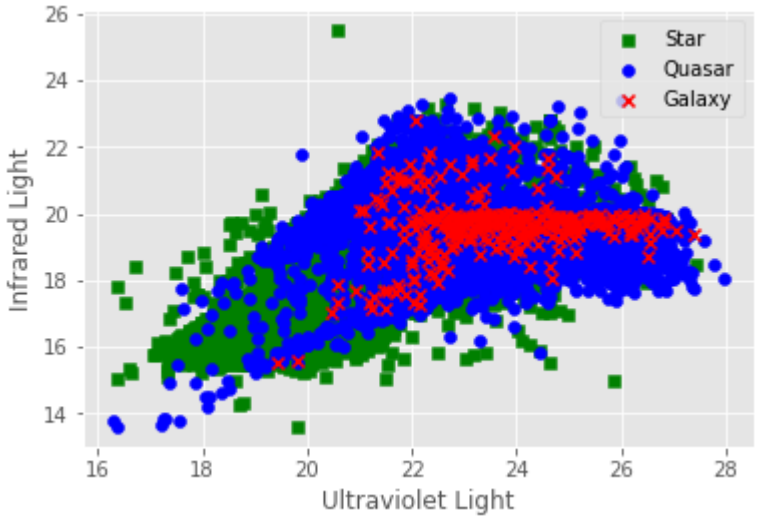


Figure 3. Scatter plot of ultraviolet, IR bands, and objects.

Differentiating objects by magnitude alone is insufficient because of the overlaps and lack of linear separability of the 3 objects. For all 3 objects, the visualization of ultraviolet vs infrared light demonstrates that all 3 classes are not linearly separable even if there is a slight shift towards a higher magnitude of UV light for galaxies. The data may need to be further partitioned based on the filter range per specific wavelength. In general, a slightly weaker and positive correlation was observed between the ultraviolet band and the rest of the bands. This observation affirms that dimensionality reduction should be implemented on highly correlated data, and any of the ML activities for the

prediction of the objects should be based on the components of Principal Component Analysis (PCA) that holds most of the variation among the features. Thus, using normalized values of 5 features, PCA was employed. We found that the first component of the PCA exhibits more than 75% of the variation (Figure 4)

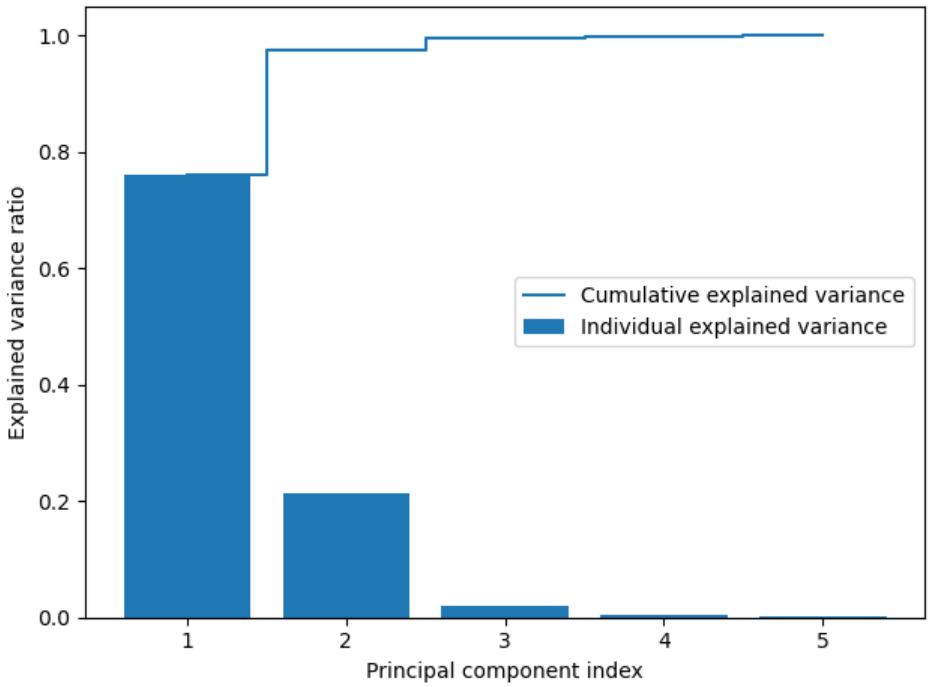


Figure 4. PCA analysis of Astronomical features. The first two components capture more than 95% of useful information.

2.3 Training and Validation before and after data reduction

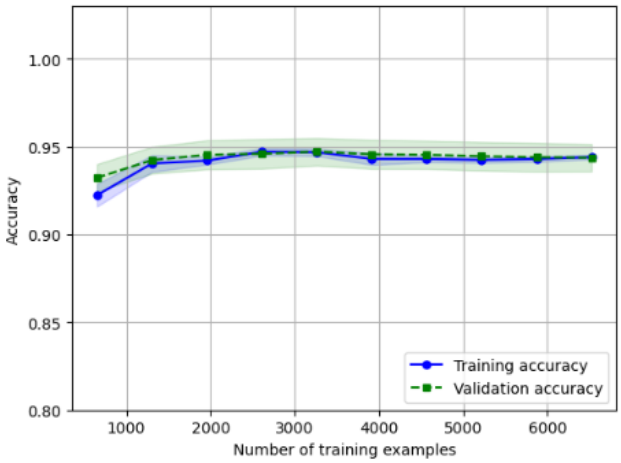
To implement the cleansed data (i.e., before dimension reduction) on the different classifiers' models, we trained our data using logistic regression, perceptron, SVC, Neighborhood, and Decision Tree classifiers. Table 2 shows the overall accuracy of classifiers before and after data reduction. The overall accuracy of each classifier indicated that the linear SVM model performed better than the rest of the ML algorithms (i.e., before reduction) (Table 2). On the other hand, the performance of classifiers after the data reduction showed lower accuracy (i.e., cross-validation, performance metrics, and overall accuracy) compared to their counterparts before the reduction of data dimensionality (Annex II).

No	Classifiers	Average accuracy (%) (Before data reduction)	Average accuracy (%) (After data reduction with PCA)
1	Perceptron	85	65
2	Logistic Regression	92	76
3	linear SVM	94	76
4	non-linear SVM	93	77
5	KNN	91	76
5	Decision Tree	77	77

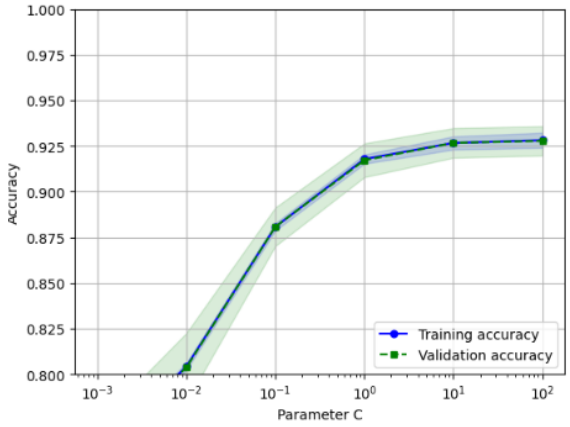
Table 2. Accuracy of initial training on common ML algorithms.

2.4 Tuning Hyperparameters

As the number of training samples in our data increases from 1000 to 3000, it also increases both training and validation accuracies (i.e., before data reduction) (figure 5A). Similarly, the C parameters obtained from logistic regression indicate that a maximum performance accuracy of identifying objects would be obtained when C is greater than 100 (figure 5B).



(B)



(A)

Figure 5. A) number of training and accuracy (training and validation accuracy) and B) parameter C from SVM

As depicted while assessing the performance of classifiers, we found that SVM exhibited higher accuracy (Table 2) – before and after data reduction. Hence, we used SVM to tune the hyperparameters using the grid search, and we found that C values of 1000 and 100, and gamma coefficient of 1 and 0.001, with the kernel definition of RBF, performed well with the accuracy of 96% (see the code cell 72) (Figure 5B)

2.5 Testing

The team conducted analysis of an additional, previously unseen, dataset to determine the accuracy of the classification. This dataset was downloaded from the SDSS Skyserver website as only “STAR” and “GALAXY” with 200,000 instances. The dataset was preprocessed in the same manner as the initial dataset used for classification and analysis. In final analysis, the initial dataset showed results of 96.06% accuracy for SVM with C = 1000 and kernel = ‘rbf’, and 94.27% accuracy for SVM with C = 100 and kernel = ‘linear’. Both sets of parameters were run against a very large unseen dataset. The SVM model with C = 100 and kernel = ‘linear’ showed superior performance of 80% vs 63% with C = 1000 and kernel = ‘rbf’ with the unseen data. Consequently, the linear kernel model is believed to be more representative of the data, rather than particular to the relatively smaller sample with which the model was originally designed upon.

As mentioned above, the accuracy for unseen data was slightly lower than the original dataset, though still a respectable 80.18%. A possible reason for this lower accuracy could be the size of the original dataset used to train the model vs the much larger testing dataset. A larger or more targeted initial dataset for the training phase may have included a more substantial number of non-”main-sequence” stars such as blue giants, red giants, and dwarf stars as shown on the Hertzsprung-Russell diagram (example found in link to the Chandra mission in section 4). These more extreme star types may be more numerous in the larger dataset and more difficult to classify.

3. Discussion and Conclusion

The curse of data dimensionality is prevalent whenever there are too many features to work with. In this project case, the curse was twofold: the reflectance of target objects could be overlapped or behaved similarly complicating the classification of target objects. On the other hand, introducing dimensionality reduction to the data might be a noise that leads to the reduction of the performance of the ML algorithms at hand. In this regard, the cross-validation results after data reduction showed lower accuracy than the cross-validation results before data reduction. However, SVM, in general, and SVM (non-linear) and SVM (linear) showed better performance, before and after data reduction, respectively.

Taking the performance evaluation after data reduction is an acceptable result, SVM (nonlinear) showed the highest performance among the rest of the ML algorithms used in this project (i.e., cross-validation accuracy 0.76 ± 0.017). The accuracy of SVM – nonlinear requires a minimum of 3000 training samples to obtain more than 90% of classification accuracy. Increasing the training or validation samples to more than 4500 might not bring any change in the accuracy of identifying astronomical objects (see Figure 5A). Similarly, we found that nonlinear SVM parameters of C=1000 and kernel of RBF are the best hyperparameter obtained via grid search with an accuracy of 96%, with a close second in C=100, kernel of linear. Finally, we concluded that the data reduction might not be necessary for the same size of data we used. Either SVM nonlinear or SVM linear ML algorithms would work for the classification of unknown data with the identified and finetuned hyperparameters obtained from the analysis.

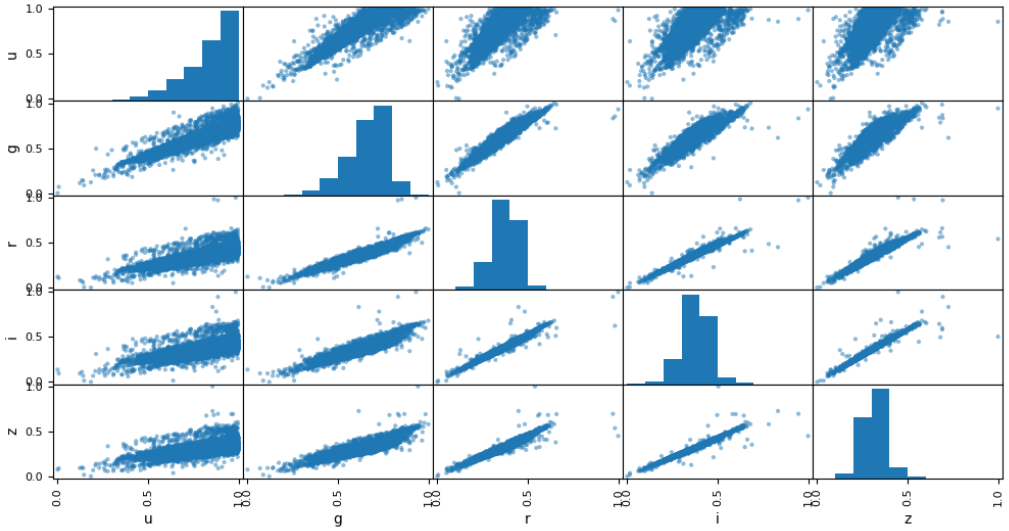
4. Related Works:

A related work on the topic is, 'Statistics, Data, Mining and Machine Learning in Astronomy' by Ivanic. The textbook is a course in modeling spectral data using density estimation, clustering, dimensionality reduction, regression, and classification Moreover, it contains python code resources related to astronomical applications, scientific articles, and initiatives who support the application of ML in astronomy and astronomical. The following resources were also used to prepare this report.

- Advances in Machine Learning and Data Mining for Astronomy, CRC Press, Taylor & Francis Group, Eds.: Michael J. Way, Jeffrey D. Scargle, Kamal M. Ali, Ashok N. Srivastava, p. 3-10
- https://www.sciencedirect.com/science/article/abs/pii/S1384107622001555?via%3Dihub
- https://academic.oup.com/mnras/article-abstract/520/1/305/6889545?redirectedFrom=PDF
- https://towardsdatascience.com/classification-of-sky-objects-with-machine-learning-be4b05816690
- Rodríguez, Rodríguez-Rodríguez, and Woo, "On the Application of Machine Learning in Astronomy and Astrophysics."
- Bailer-Jones, "Applications of Machine Learning in Astronomy."
- Example of Hertzsprung-Russell diagram: https://chandra.harvard.edu/edu/formal/stellar_ev/story/index3.html

Annex

A. Scatter matrix plot of Bands



B. Performance metrics of classifier

Before Data Reduction

==========

Precision, Recall, F1-score, and Support before data reduction

==========

Model Score of perceptron:

Accuracy: 0.89 1813

	precision	recall	f1-score	support
0	0.96	0.85	0.90	1084
1	0.81	0.95	0.87	729
macro avg	0.88	0.90	0.89	1813
weighted avg	0.90	0.89	0.89	1813

Model Score of Logistic Regression:

Accuracy: 0.92 1813

	precision	recall	f1-score	support
0	0.94	0.91	0.92	979
1	0.90	0.93	0.91	834
macro avg	0.92	0.92	0.92	1813
weighted avg	0.92	0.92	0.92	1813

Model Score of SVM (non-linear):

Accuracy: 0.93 1813

	precision	recall	f1-score	support
0	0.94	0.94	0.94	958
1	0.93	0.93	0.93	855
macro avg	0.93	0.93	0.93	1813
weighted avg	0.93	0.93	0.93	1813

Model Score of SVM (linear):
Accuracy: 0.95 1813

	precision	recall	f1-score	support
0	0.95	0.95	0.95	955
1	0.95	0.94	0.94	858
macro avg	0.95	0.95	0.95	1813
weighted avg	0.95	0.95	0.95	1813

Model Score of KNN:
Accuracy: 0.92 1813

	precision	recall	f1-score	support
0	0.94	0.91	0.92	985
1	0.89	0.93	0.91	828
macro avg	0.91	0.92	0.92	1813
weighted avg	0.92	0.92	0.92	1813

Model Score of Decision Tree classifier:
Accuracy: 0.78 1813

	precision	recall	f1-score	support
0	0.84	0.76	0.80	1057
1	0.70	0.80	0.75	756
macro avg	0.77	0.78	0.77	1813
weighted avg	0.78	0.78	0.78	1813

After Data Reduction (PCA)

==========
Precision, Recall, F1-score, and Support after data reduction
==========

Perceptron After PCA Data reduction
Accuracy: 0.73 1813

	precision	recall	f1-score	support
0	0.72	0.75	0.73	913
1	0.74	0.70	0.72	900
macro avg	0.73	0.73	0.73	1813
weighted avg	0.73	0.73	0.72	1813

Linear Regression After PCA Data reduction

Accuracy: 0.75 1813

	precision	recall	f1-score	support
0	0.81	0.74	0.77	1052
1	0.68	0.77	0.72	761
macro avg	0.75	0.75	0.75	1813
weighted avg	0.76	0.75	0.75	1813

SVM After PCA Data reduction
Accuracy: 0.76 1813

	precision	recall	f1-score	support
0	0.82	0.75	0.79	1046
1	0.70	0.78	0.74	767
macro avg	0.76	0.77	0.76	1813
weighted avg	0.77	0.76	0.77	1813

Linear SVM linear After PCA Data reduction
Accuracy: 0.75 1813

	precision	recall	f1-score	support
0	0.79	0.75	0.77	1015
1	0.70	0.75	0.73	798
macro avg	0.75	0.75	0.75	1813
weighted avg	0.75	0.75	0.75	1813

KNN After PCA Data reduction
Accuracy: 0.75 1813

	precision	recall	f1-score	support
0	0.84	0.73	0.78	1091
1	0.66	0.78	0.72	722
macro avg	0.75	0.76	0.75	1813
weighted avg	0.77	0.75	0.75	1813

Decision tree After PCA Data reduction
Accuracy: 0.75 1813

	precision	recall	f1-score	support
0	0.85	0.72	0.78	1126
1	0.64	0.80	0.71	687
macro avg	0.75	0.76	0.75	1813
weighted avg	0.77	0.75	0.76	1813

Runtimes for processes (hh:mm:ss.ss)

Runtime for entire program	0:20:39.94
Runtime for first model fitting/testing (All models)	0:00:19.09

PCA runtime	0:05:01.85
Cross Validation runtime	0:03:25.91
Runtime for grid search	0:11:09.16
Runtime to fit initial data	0:00:04.09
Runtime to test the 200k dataset	0:00:04.85