Kidney Stone Prediction based on Urine Analysis

Jerin Ishrat Natasha, ID:40221272

GitHub link: https://github.com/jerin00/INSE6220_40221272.git

Abstract—***Principal component analysis (PCA) is a method for reducing the dimensionality of large datasets which increases interpretability but at the same time minimizes information loss while preserving most of the information in the data. In this report PCA is applied on Kidney Stone Prediction based on Urine Analysis to detect the presence of Stones formed in kidneys. Three different classification algorithms; i.e; logistic regression (LR), Gradient Boosting (GB), and Decision Tree (DT) are applied on original dataset and transformed dataset (after applying PCA) to identify the presence or absence of kidney stones. In the next stage each model is tuned with ideal hyperparameters to achieve better performance metrics and the performance of each algorithm is found using F1 score, confusion matrix and receiver operating characteristic (ROC) curves. DT shows the best performance for the dataset among all other available machine learning models in PyCaret library. Next, an experiment is for the interpretation of the model is shown using the explainable AI (artificial intelligence) Shapley values. Extra trees (ET) classifier model has been used for this purpose. The report shows how the algorithms successfully determines the two classes of and F1-score nearly 1 is obtained.***

Index terms—Principal component analysis, binary classification, logistic regression, Gradient boosting, Decision Tree

I. INTRODUCTION

"Kidney stones are hard, non-organic deposits which are formed inside kidneys when a person's excrement is concentrated and contain more transparent making ingredients than the fluid in his urine". Some of these elements include Uric acid, calcium, and oxalate. Urinary calculus is developed when our bodies get dehydrated but have a lot of waste. This has been a major problem in recent years, and it can lead to vital health risks if not diagnosed on time. It can even cause the necessity for surgery to remove the stone.[1]

Using urine analysis as a guide for the diagnosis and treatment of kidney stones has been recommended for many stone formers in all of the published international guidelines [2–5], but data suggests that it is not usually utilized as widely as it has been recommended. For instance, a recent study of a large cohort within the United States (US) Veterans Affairs Health Care System found out that less than 1 out of 6 stone forming patients had taken 24 hours urine testing that would have been relevant to managing their urinary stone disease [6]. One possible explanation of this low utilization of urine data could be that physicians in the US are not fully convinced that urine testing is valuable and can be cost effective [7–9]. In the UK, the majority of the health authorities have stopped the routine biochemical screening of stone patients in order to save money and manage kidney stone patients by only the urological removal/disintegration of the stones. However, this method does not "cure" the patients' risk of forming further stones in them. It is seen that this strategy actually costs even more than would be the case if proper biochemical screening were applied and thus resulting in a reduction in stone recurrence [10].

In recent years, ML techniques have proven to play a significant role in the diagnosis of Kidney Stone by applying classification techniques to identify people with kidney stones based on Urine Analysis. In this report, at first Principal component analysis (PCA) is applied on the Kidney Stone Prediction based on Urine Analysis dataset aiming to reduce dimensionality. Then, three popular classification algorithms, logistic regression (LR), Decision Tree (DT) and Gradient Boosting Classifier are applied on the original dataset and the PCA transformed dataset. The purpose is to determine whether a patient has kidney stones or not. Finally, classification models are interpreted using explainable AI (artificial intelligence) with Shapley values. In this report, the presented results from the classification algorithms represent the results obtained after applying PCA; that is, the results are used from the transformed dataset. The classification results of the original dataset can be found on the Google Colab notebook. The rest of the report is organized as: Section II. describes the PCA methodology, Section III. provides an overview of the three classification algorithms, Section IV provides the Kidney Stone Prediction based on Urine Analysis dataset description, Section V discusses about the PCA results, Section VI gives an extensive analysis of the classification results, Section VII discusses on the explainable AI Shapley values and in section VIII, the report is concluded.

II. PRINCIPAL COMPONENT ANALYSIS

Large datasets are largely common in many disciplines but are often difficult to interpret. Principal component analysis (PCA) is a method for reducing the dimensionality of such large datasets, increasing interpretability but at the same time minimizing information loss and preserving most of the information in the data. PCA does so by creating new uncorrelated variables that can successively maximize the variance.

A. PCA Algorithm

The PCA algorithm is applied to a data matrix X with dimension n × p using the followings steps [11]:

1. **Standardization**- In this step, the range of continuous initial variables is standardized to analyze the contribution of each variable equally. To do this, compute the mean vector $\bar{x}$ of each column of the data set. The mean vector is a p dimensional vector which can be found by:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

The data can be standardized by subtracting the mean of each column from each item in the data matrix. The final centered data matrix (Y) can be expressed as:

$$Y = HX, \quad (2)$$

where H represents the centering matrix

2. **Covariance matrix computation:** This step figures out how the variables of the given data are varying with the mean value calculated. Variables can be too highly correlated because of which it might contain redundant information. At the end of this step, any interrelated variables can also be sorted out. To distinguish the highly interrelated variables, the p x p covariance matrix is calculated with the formula below-

$$S = \frac{1}{n-1} Y^T Y. \quad (3)$$

3. **Performing eigen decomposition:** The eigenvectors indicate the principal components of the covariance matrix and the eigenvalues are their corresponding magnitude. The eigenvector with the largest corresponding eigenvalue represents the direction of the maximum variance. Eigen decomposition can be calculated using the following equation:

$$S = A\Lambda A^T \quad (4)$$

Where A is the p × p orthogonal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues.

4. **Principal components:** This step computes the transformed matrix Z of size n×p. The rows of Z are the observations and columns are the PCs; the number of PCs is equal to the dimension of the original data matrix. Z can be calculated by:

$$Z = YA \quad (5)$$

III. MACHINE LEARNING BASED CLASSIFICATION ALGORTHIMS

A. Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method which are used for classification and regression. The goal of this algorithm is to create a model that can predict the value or class of a target variable by learning simple decision rules that are inferred from the prior data (training data).

Decision Trees, for predicting a class label for a record, starts from the root of the tree. Then the values of the root attribute are compared with the record's attribute. Based on this comparison, the branch corresponding to that value is followed and jumps to the next node.

Some of its advantages include-

- It is simple to understand and to interpret and trees can be visualized.
- It requires little data preparation, while other techniques often require data normalization and the need for creation of dummy variables.

However, decision-tree learners can often create over-complex trees that do not generalize the data well. This is known as overfitting. Another drawback is that decision trees can be unstable because small variations in the data can result in a completely different tree being generated. But this problem can be mitigated by using decision trees within an ensemble.

B. Logistic Regression

Logistic Regression Algorithm is a machine learning technique used for predicting the categorical dependent variable using a given set of independent variables. It creates the best fitting model for the dataset to develop a relationship between the class and its features. It labels the samples as 1 or 0.

The logistic function is represented by-

$$S(z) = \frac{1}{1+e^{-z}} \quad (6)$$

Where, the output of S(z) lies between 0 and 1 and z = the input to the function.

One of the benefits of using Logistic regression is it is easier to implement, interpret, and very efficient to train in the Kidney Stone Prediction classifier problem to state whether a patient has Kidney Stones or not. But one of the drawbacks of this algorithm is that if the number of observations is less than the number of features, Logistic Regression should not be used because it may cause overfitting. The major disadvantage of Logistic

Regression is that it assumes linearity between the dependent and the independent variables.

C. Gradient Boosting

The Gradient Boosting algorithm is one of the most powerful algorithms in the field of machine learning. It is known that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. Since gradient boosting is one of the boosting algorithms, it is used to minimize bias error of a model.

Gradient boosting algorithms can be used for predicting not only continuous target variables (as a Regressor) but also categorical target variables (as a Classifier) as has been used on the Kidney Stone Prediction dataset.

IV. DATASET DESCRIPTION

The Kidney Stone Prediction based on Urine Analysis dataset used for this project has been collected from Kaggle. The dataset provides information about the presence or absence of stones in a patient. The dataset projects 6 features for the detection of kidney stones. The features are: “gravity” (specific gravity of urine), “ph” (ph of urine), “osmo” (osmolarity of urine), “cond” (conductivity of urine), “urea” (concentration of urea in urine), “calc” (concentration of calcium in urine). It includes 79 entries for each of these attributes. Finally, it contains a column titled “target” which represents the label for the class to identify the presence or absence of stones 0 represents absence of stone and 1 represents presence of stone.

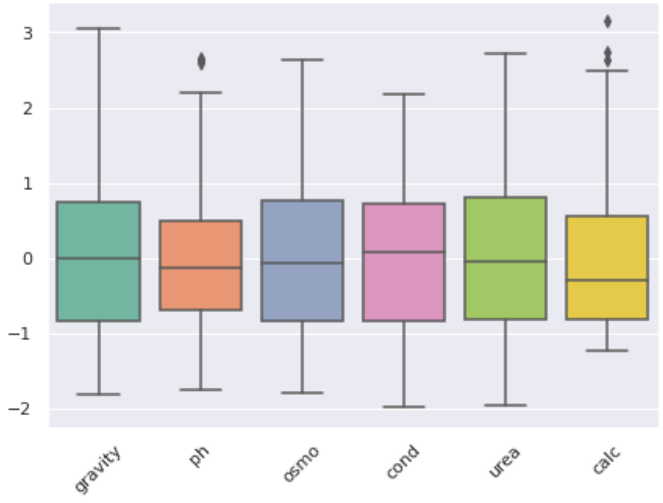


Fig. 1. Box Plot

With the help of the box and whisker plots and their five number summaries on the dataset, the distributions, central values and variability of the features were measured. Fig. 1 illustrates the box plot of the features of the kidney stone dataset. Fig. 1 indicates that most of the features follow approximately normal distribution. However, outliers exist in two features: “ph” and “calc”. The outliers in these two features lie on the left.

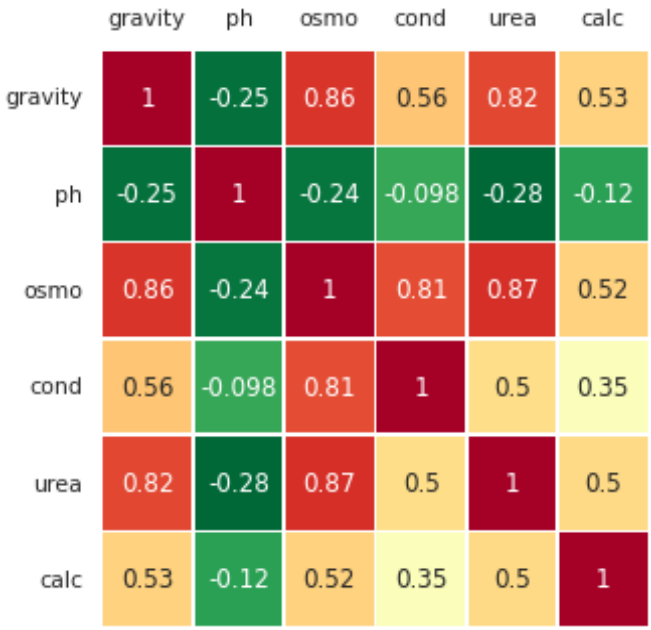


Fig. 2. Correlation matrix

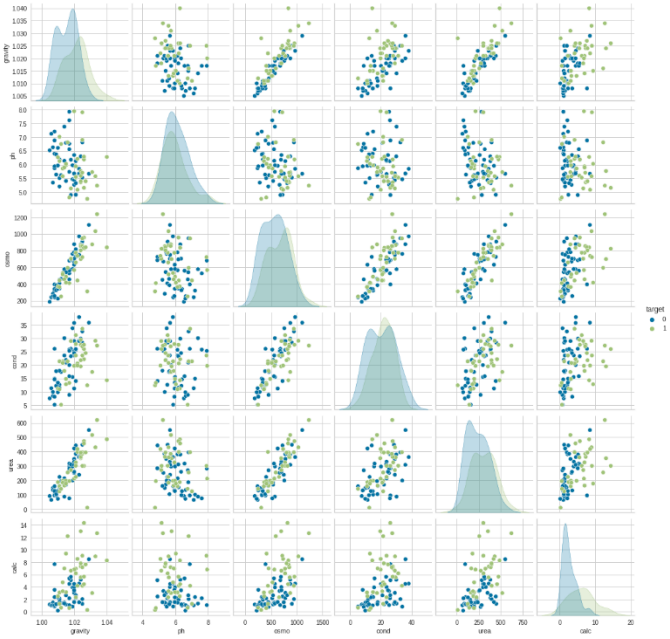


Fig. 3. Pair plot

Fig. 4. Fig. 2 shows the correlation matrix for the normalized features of the kidney stone dataset. The features with large positive numbers are gravity, osmo and urea. This indicates that these three features are highly correlated. Other three features; i.e. ph, cond and calc show less correlation with the other features in the dataset. The observation can be seen in Fig. 3 illustrating the pair plot. The highly correlated features gravity,

osmo and urea contain higher numbers of cells with regularly increasing lines. However, ph, cond and calc illustrates less apparent correlation.

V. PCA RESULTS

In this project, PCA has been applied on the kidney stone dataset by implementing PCA using the PCA library. By implementing the PCA steps, the feature set of 6 can be reduced to r numbers of features where r < 5. The original n × p dataset can be reduced using eigenvector matrix A where each column of the eigenvector matrix A is represented by a PC. Each PC takes an amount of data from the dataset that determines the dimension (r). The obtained eigenvector matrix (A) for kidney stone dataset is as follows:

$$A = \begin{bmatrix} 0.474 & -0.007 & 0.035 & 0.369 & 0.779 & 0.173 \\ -0.169 & 0.950 & 0.072 & 0.254 & -0.010 & -0.008 \\ 0.50 & 0.090 & -0.204 & 0.029 & -0.131 & -0.821 \\ 0.392 & 0.270 & -0.526 & -0.589 & -0.019 & 0.385 \\ 0.467 & -0.059 & 0.056 & 0.504 & -0.611 & 0.384 \\ 0.341 & 0.117 & 0.820 & -0.444 & -0.035 & 0.010 \end{bmatrix}$$

and the corresponding eigenvalues are:

$$\lambda = \begin{matrix} 3.711 \\ 0.966 \\ 0.700 \\ 0.499 \\ 0.178 \\ 0.022 \end{matrix}$$

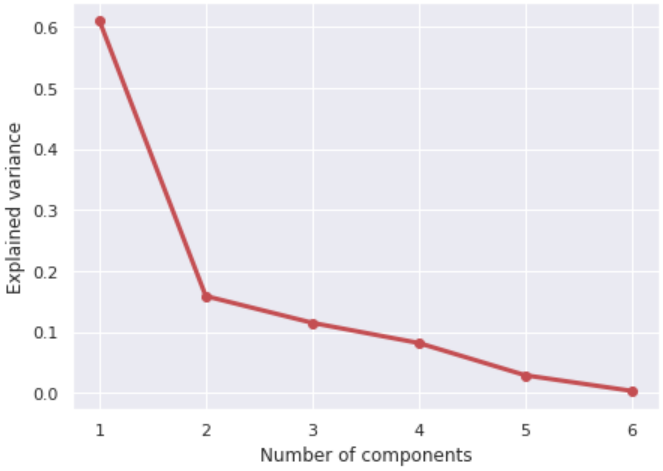


Fig. 4. Scree Plot

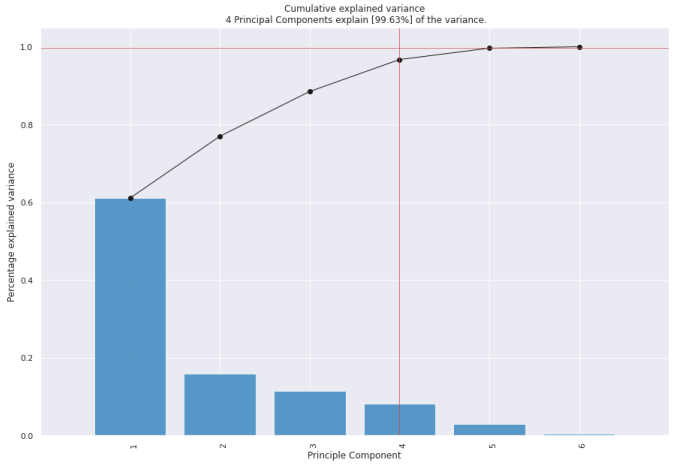


Fig. 5. Pareto Plot

Fig 4 and Fig. 5 demonstrate the scree plot and pareto plot of the PCs respectively. The scree plot and pareto plot display the amount of variance explained by each principal component. The percentage of variance experienced by j-th PC can be evaluated using the following equation:

$$j = \frac{\lambda_j}{\sum_j^p \lambda_j} \times 100, j = 1, 2, \dots, p, \quad (7)$$

Where λ_j is the eigenvalue and the amount of variance of the j-the PC. It can be observed from the two figures that the variance of first three PC's contribute to 89.0% of the amount of variance of the original dataset; i.e. first PC holds 60.4% of variance (l1= 61.1%), the second PC holds 15.9% of variance (l2 = 16.4%) and the third PC holds 11.5% of variance (l3 = 11.5%), combining to a total of 89.0%. The scree plot presents that the elbow is located on the second PC. These two observations imply that the dimension of the feature set can be reduced to two (r = 2). The first principal component Z1 is given by:

Z1 = 0.474X1 -0.169X2 + 0.508X3 + 0.392X4 + 0.467X5+ 0.341X6 (8)

It can be observed from the first PC that X1 (gravity), X3 (osmo), X5 (urea) contribute the most to the first PC. But none of the features have a negligible contribution to the first PC. The second principal component Z2 is given by:

Z2 = −0.007X1 +0.950X2 + 0.090X3 + 0.270X4 - 0.059X5 +0.117X6 (9)

From the second PC Z2 it can be seen that X2 (ph), X4(cond) and X6 (calc) have the highest contribution in the second PC. The contributions for X1 (gravity), X3 (osmo), X5 (urea) are very small and hence negligible. Therefore, Z2 can be rewritten as follows:

Z2 = 0.950X2 + 0.270X4 +0.117X6 (10)

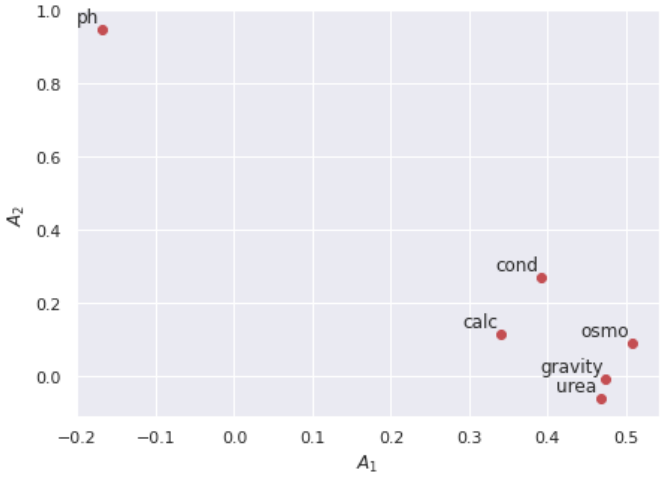


Fig. 6. PC Coefficient Plot

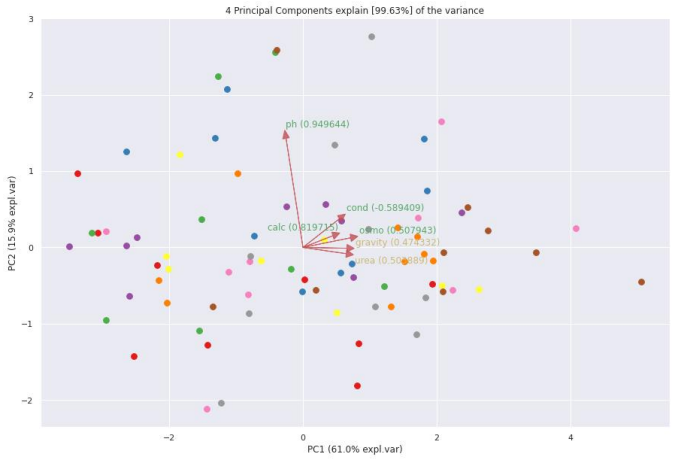


Fig. 7. PC Coefficient Plot

Fig. 6 shows the PC coefficient plot. It represents the amount of contribution each feature of the dataset has on the first two PCs. It can be seen that the figure supports the previous calculation of PCs and gravity, osmo and urea has the highest contributions in the first PC. On the other hand, ph, cond and calc have the greatest contribution to the second PC. The 'urea' (location=(0.47,-0.05)) has the only negative coefficient and is located at the bottom right side of the plot and far away from Fig. 4: Scree Plot Fig. 5: Pareto Plot the clusters of features on the right side of the graph. "ph" is located far away from the other features.

The Biplot in Fig. 7 illustrates a different visual representation of the first two PCs. The two axes of the biplot represent the first two PCs and the rows of the eigenvector matrix are shown as a vector. Each of the observations in the kidney stone dataset is drawn as a dot on the biplot. The vectors for features- gravity, osmo and urea show very small angles with the first PC and very large angles with the second PC. This observation supports the analysis of the PC coefficient plot of Fig. 6. It indicates that these three features have a large contribution to the first PC and a very small contribution to the second PC. On the other hand, the vectors for ph, cond, calc show the opposite observation. They create a bigger angle with the first PC and smaller angle with the second PC. This implies that they are more related to the second PC rather than the first PC. Additionally, the vectors that follow the same direction are positively correlated with each other. For example, cond, calc, osmo, gravity and urea are facing in the same direction.

VI. CLASSIFICATION RESULTS

This section focuses on the performance of three popular classification algorithms on the kidney stone prediction based on urine analysis dataset. To identify the effects of PCA on the dataset, the classification algorithms are applied on the original dataset as well as the PCA applied dataset with three PCA components. In order to perform the classification, the PyCaret library of Python has been used. The original dataset has been split into a train and test set with the proportion of 70% and 30%, respectively. For reproducibility, the session id has been set with 123.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.860	0.8417	0.8167	0.9100	0.8383	0.6873	0.7058	0.093
ridge	Ridge Classifier	0.840	0.0000	0.7667	0.8100	0.7717	0.6328	0.6446	0.009
gbc	Gradient Boosting Classifier	0.840	0.8667	0.8167	0.8750	0.8257	0.6649	0.6837	0.048
lr	Logistic Regression	0.820	0.8833	0.7333	0.9100	0.7850	0.6034	0.6337	0.263
ada	Ada Boost Classifier	0.820	0.8917	0.7667	0.7583	0.7490	0.6040	0.6225	0.070
rf	Random Forest Classifier	0.815	0.8750	0.7333	0.7917	0.7424	0.5988	0.6302	0.186
lda	Linear Discriminant Analysis	0.800	0.8667	0.7333	0.7767	0.7317	0.5558	0.5779	0.010
et	Extra Trees Classifier	0.795	0.8750	0.6833	0.7917	0.7090	0.5534	0.5914	0.134
dt	Decision Tree Classifier	0.775	0.7750	0.8667	0.7333	0.7781	0.5441	0.5877	0.010
nb	Naive Bayes	0.750	0.9167	0.7667	0.7100	0.7098	0.4418	0.4670	0.009
qda	Quadratic Discriminant Analysis	0.715	0.8500	0.7500	0.7183	0.6840	0.4029	0.4544	0.011
knn	K Neighbors Classifier	0.550	0.6000	0.6667	0.5867	0.5900	0.0712	0.0779	0.014
svm	SVM - Linear Kernel	0.490	0.0000	0.3000	0.1700	0.2167	-0.0506	-0.0612	0.009
dummy	Dummy Classifier	0.410	0.5000	0.5000	0.2100	0.2952	0.0000	0.0000	0.009

Fig. 8. Comparison among classification models before applying PCA

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.840	0.8250	0.7333	0.9250	0.7957	0.6579	0.6949	0.011
gbc	Gradient Boosting Classifier	0.775	0.8500	0.6833	0.7917	0.7090	0.5170	0.5506	0.047
lr	Logistic Regression	0.770	0.9000	0.7667	0.7850	0.7407	0.4888	0.5096	0.011
qda	Quadratic Discriminant Analysis	0.755	0.8500	0.6500	0.7167	0.6481	0.4674	0.5081	0.011
svm	SVM - Linear Kernel	0.750	0.0000	0.7333	0.7667	0.7114	0.4664	0.5041	0.012
ridge	Ridge Classifier	0.750	0.0000	0.6833	0.7000	0.6648	0.4503	0.4762	0.009
lda	Linear Discriminant Analysis	0.750	0.9000	0.6833	0.7000	0.6648	0.4503	0.4762	0.010
nb	Naive Bayes	0.735	0.8667	0.6667	0.7667	0.6681	0.4511	0.4989	0.010
rf	Random Forest Classifier	0.730	0.8167	0.7000	0.8000	0.7014	0.4416	0.4858	0.311
knn	K Neighbors Classifier	0.715	0.7667	0.6500	0.7183	0.6407	0.3755	0.4156	0.017
ada	Ada Boost Classifier	0.715	0.8083	0.6500	0.7250	0.6524	0.4141	0.4469	0.134
et	Extra Trees Classifier	0.710	0.8667	0.6500	0.7500	0.6514	0.3886	0.4283	0.145
lightgbm	Light Gradient Boosting Machine	0.670	0.7250	0.7000	0.6867	0.6500	0.2801	0.2946	0.015
dummy	Dummy Classifier	0.410	0.5000	0.5000	0.2100	0.2952	0.0000	0.0000	0.010

Fig. 9. Comparison among classification models after applying PCA

PyCaret helps to create a performance comparison table among all available classification algorithms based on the target dataset and identify the best model that has the highest accuracy. It can be observed from the Fig. 8 that, before applying PCA, the best three classification models with the highest accuracies on kidney stone dataset are Light Gradient Boosting Machine, Ridge

Classifier and Gradient Boosting Classifier. However, in Fig. 9 the comparison of the classification models after applying PCA is seen to be different. In this case, the best three models which give the highest accuracy on the transformed dataset are Decision Tree Classifier, Gradient Boosting Classifier and Logistic Regression. So, these three algorithms have been considered for the evaluation purposes during the rest of the experiment. The original and transformed dataset have been trained, tuned and evaluated with these three algorithms. Both of the experiments (classification algorithms applied on original dataset and transformed dataset) have been presented in Google Colab notebook although this report only focuses on the results obtained after the application of PCA (transformed dataset).

```
[286] tuned_dt_pca = tune_model(dt_pca)
```

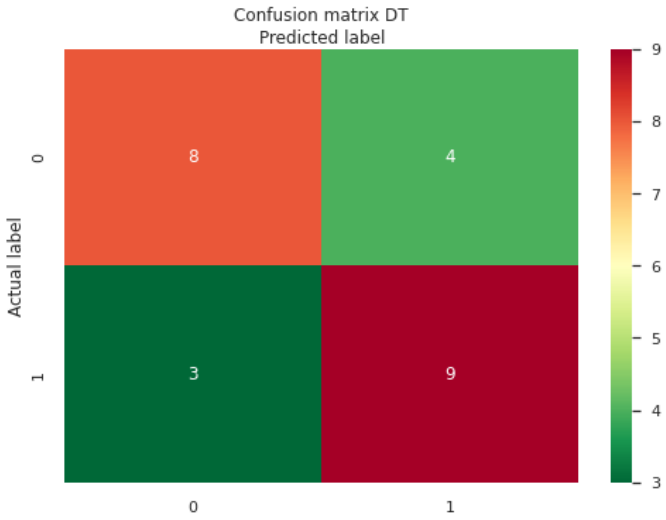
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6000	0.1667	0.0000	0.0	0.0000	0.0000	0.0000
1	0.8000	0.8333	0.5000	1.0	0.6667	0.5455	0.6124
2	0.8000	0.5000	0.5000	1.0	0.6667	0.5455	0.6124
3	0.8000	0.9167	0.5000	1.0	0.6667	0.5455	0.6124
4	0.8000	0.7500	0.6667	1.0	0.8000	0.6154	0.6667
5	1.0000	1.0000	1.0000	1.0	1.0000	1.0000	1.0000
6	1.0000	1.0000	1.0000	1.0	1.0000	1.0000	1.0000
7	0.8000	0.9167	0.6667	1.0	0.8000	0.6154	0.6667
8	1.0000	1.0000	1.0000	1.0	1.0000	1.0000	1.0000
9	0.7500	0.5000	0.5000	1.0	0.6667	0.5000	0.5774
Mean	0.8350	0.7583	0.6333	0.9	0.7267	0.6367	0.6748
SD	0.1226	0.2673	0.2963	0.3	0.2788	0.2908	0.2818

Fig. 10. DT metrics score after hyparameter tuning

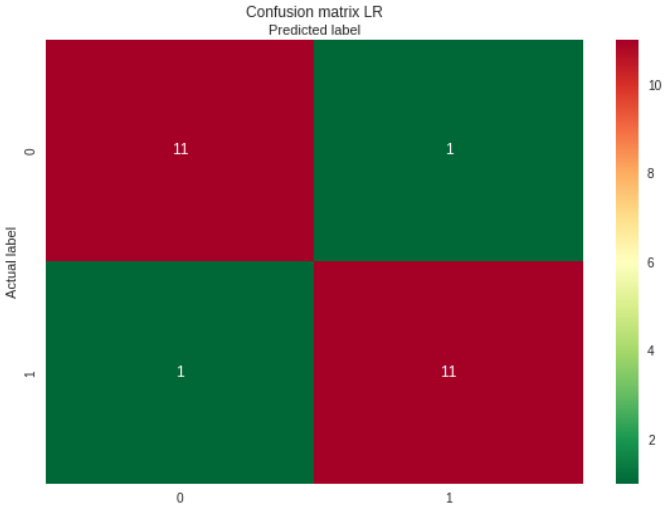
When creating a machine learning model, there are many design choices as to how to define our model architecture. Quite often we can't instantly figure out what the optimal model architecture should be for a given model, and thus we have to explore a range of possibilities. With the help of machine learning, we can ask the machine to perform this exploration and choose the optimal model architecture automatically. Parameters that define the model architecture are called hyperparameters and this process of searching for the ideal model architecture is known as hyperparameter tuning. Hyperparameter tuning can be really beneficial to improve the performance of a model. Hyperparameter tuning with PyCaret is done in three steps; at first creating a simple baseline model, then tuning the model and finally evaluating its performance.

First, a classification model per algorithm is generated. Next, the tune_model() function is used for tuning the model with the ideal hyperparameters. The role of this function is to automatically tune the model with effective hyperparameters that work on a pre-defined search space and score it with the help of stratified K-fold cross validation. By default, PyCaret can apply 10 folds stratified K-fold validation on the three algorithms.

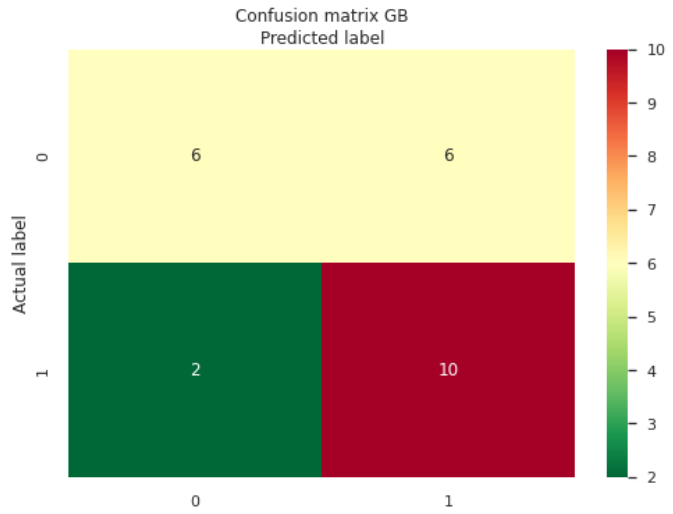
Moreover, the LR model is tuned with L2 penalty which is a regularization technique to prevent overfitting problems. For DT, the criterion "gini" is tuned and the "entropy" criterion for regularization is tuned for Gradient Boosting. Fig. 10 indicates that tuned LR model metrics work better than the base model metrics (before hyperparameter tuning).



A) Decision Tree



B) Linear Regression



C) Gradient Boosting Classifier

Fig. 11. Confusion matrices of the three classification algorithms applied on transformed dataset

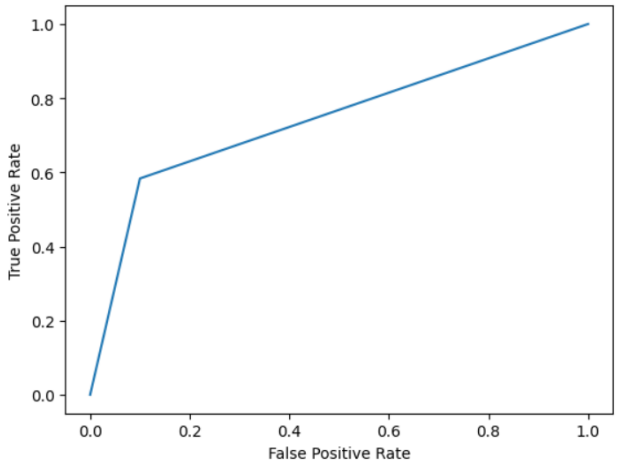
Precision and Recall are a useful measure of success of prediction when the classes are very imbalanced. In case of information retrieval, precision is a measure of result relevancy, and recall is a measure of how many truly relevant results are returned.[12] As the kidney stone dataset is a binary classification problem, the precision and recall measurements can evaluate the performance of each class individually. The obtained results from precision and recall are shown using the confusion matrices in Fig. 11. The confusion matrix represents the matrix that provides the mix of predicted vs. the actual class instances. It shows the correct and incorrect predictions with count values and breaks them down for every class. In the Fig. 11 the confusion matrix tables for the three algorithms that were applied on the transformed dataset are illustrated. The confusion matrices for the original dataset are present in the Google Colab notebook. In the figure, the horizontal axis represents the predicted label and vertical axis represents the actual label. DT misclassified 4 instances from class 0 (absent) as class 1 (present) and 3 instances of class 1 (present) are misclassified as class 0 (absent). LR misclassified 1 instance of class 0 (absent) and 1 instance of class 1 (present) whereas GB misclassified 6 instances of class 0 (absent) and 2 instances of class 1 (present).

The performance evaluation can also be measured as F1-score. The F1-score aims to combine the precision and recall of a classifier and turn them into a single metric by using their harmonic mean. It is a great metric when it comes to comparing the results among the classifiers. For instance: classifier A has a higher recall, and classifier B has a higher precision. In such cases, the F1-score helps to identify the better classifier. The function of F1- score can be defined as below:

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

It can be observed from Fig. 8 and Fig. 9 that the F1-score of DT, LR, and Gradient Boosting has improved significantly after applying PCA. So, it is evident that the dimension reduction weakens the dependencies among the features of the dataset. F1- score is enhanced even more after the model is tuned with its ideal hyperparameters. Thus, it shows that application of PCA and hyperparameters tuning is very beneficial.

A)



B)

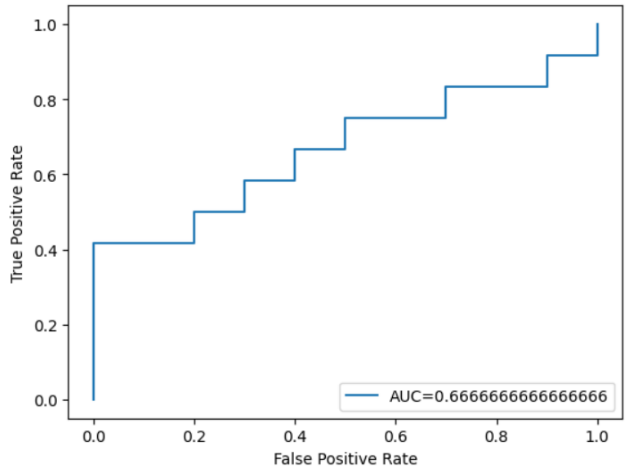


Fig. 12. (A and B) – ROC curve Decision tree

In the final step of analysis, the receiver operating characteristic (ROC) curve for DT algorithm can be seen in Fig. 12. ROC curve aims to demonstrate the performance of a

classification model at every classification threshold. The ROC curve consists of two parameters- True Positive Rate and False Positive Rate which build the confusion matrix.So, ROC curve and confusion matrix are very closely related and show different visual representation of the same measurement. ROC curves for LR and GBC have been shown in the Google Colab notebook. The ROC curve of DT can be seen in Fig. 12 which reflects the results of the confusion matrix. The graph plots the false positive rate on the x-axis and the true positive rate on the y-axis for the different candidate threshold values between 0.0 and 1.0. It also illustrates the graph of macro and micro average curve. The ROC curve and AUC values shows that LR is the best at predicting both classes and can predict 75% accurately. So, it is observed that the three algorithms can effectively classify the presence or absence of kidney stone.

VII. EXPLAINABLE AI WITH SHAPLEY VALUES

Model interpretability is the extent to which a human can understand the cause of a decision and consistently predict the model's result. Therefore, it is an important metric in the context of ML. At a high level, the aim of the Shapley value approach is to explain why an ML model reports the outputs that it does on an input. There are several ways of improving the interpretability of a mode, for instance, feature importance. Feature importance enables us to estimate the contribution of every feature in the prediction process. So, to get an overview of the most important features on the PCs, the SHAP values are used. This is done by importing the open source "shap" library in Python. Shapley values are a concept borrowed from the cooperative game theory literature and date back to the 1950s. Shapley values refers to the contribution of each feature in pushing the prediction away from the expected value. [13]

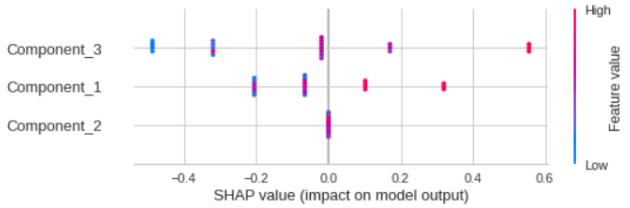


Fig. 13. Summary Plot

Fig. 14. Force plot for a single observation

A player can either be an individual feature value, for instance. for tabular data. or a group of feature values. Shapley values focuses on how to adequately distribute the prediction among the feature set. It is to be noted that the shap library of Python is still in its development stage and therefore it only supports tree-based models, such as, decision tree, random forest, extra trees classifier, etc. for binary classification problems. Since, kidney stone diagnosis dataset is a binary classification problem, the shap analysis cannot be run on LR and Gradient Boosting. So, the shap analysis has been performed on "Decision Tree Classifier ". First of all, a DT model is created and tuned with ideal hyperparameters. Next, the tuned model is passed to the shap library to generate the interpretation plots. In my case, each of the PCs acts as a player in the coalition.

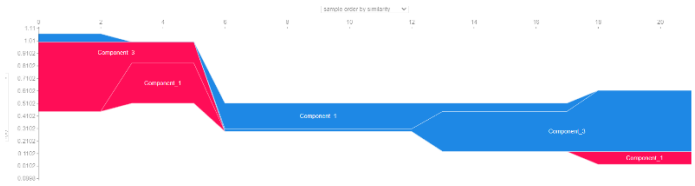


Fig. 15. Combined Force Plot

Fig. 13 illustrates the summary plot of SHAP values. Here, the summary plot combines feature importance along feature effects. Each of the points on the summary plot corresponds to the Shapley value for a PC and an instance. The y-axis of the plot represents the PCs and the x-axis corresponds to the Shapley values. Component_1 is the first PC, component_2 is the second PC and component_3 is the third PC. The PC's are arranged according to their importance. Thus it is evident that the pareto plot and scree plot indicate that the first PC carries the most feature variance. The red color represents high PC value and blue color represents low PC value. From the observation of the summary plot, it can be said that a low level of PC value has a high and positive impact on the kidney stone diagnosis. And, a high level of PC value has a low and negative impact on the kidney stone diagnosis. Specifically, PCs are negatively correlated with the target variable. Fig. 14 shows the force plot for a single observation. In this example, the 32nd observation is chosen. This plot illustrates how each of the features contributes to pushing the model output from the base value. Here, the base value indicates the value that would be predicted if there were no features known for the current output. That is, it is the mean prediction of the test set. Here, the base value is 0.5102. In the plot, the bold value 0.29 represents the model's score for this observation. Higher scores drive the model to predict 1 and lower scores drive the model to predict 0. The blue color on the first and third PC implies that they are pushing the prediction to be lower. This particular observation is classified as class 0 (absent) because it is pushed more to the left. But, this plot is only an output for this observation and does not describe the predicted output of the entire model. In Fig.15 the combined force plot of all PCs can be seen. This plot combines all individual force plots with a 90 degree rotation and are stacked horizontally on each other. In this plot, it can be seen that the y-axis is the x-axis of the individual force plot. There are 22 data points in the

transformed test set, hence the x-axis has 22 observations. This combined force plot indicates the effect of each PC on the current prediction. Values in the blue colour represents a positive influence on the prediction whereas values in the red colour represents a negative influence on the prediction.

VIII. CONCLUSION

To conclude, PCA and three popular classification algorithms have been applied on kidney stone prediction based on Urine Analysis dataset. The dataset contains information on attributes to identify the presence or absence of kidney stones. At first, PCA has been applied to the original dataset. The first three PC's correspond to 89% variance of the data. Hence, the featureset is reduced to 2 from 5. Multiple experiments have been carried out on the first two PC's and different plots have been generated in order to verify the obtained results from different perspectives. Next, three classification algorithms, LR, DT and Gradient Boosting have been applied on the original dataset and transformed dataset with the first three components. The three algorithms have been tuned with the ideal hyperparameter settings and performance evaluation was conducted by comparing the different confusion matrices, ROC curves and F1-scores. It can be seen that after hyperparameter tuning performance metrics score of each algorithm has improved noticeably. The LightGBM, Ridge and GBC algorithms performed the best on the original dataset. However, after applying PCA, DT, LR and GBC performed the best and generated the best performance metrics. Finally, to improve the interpretability of the model, multiple interpretation plots have been generated using explainable AI shapley values. Overall, all three algorithms have proved to effectively determine the tumor types for kidney stone diagnosis.

REFERENCES

[1] Vishnu Prasad G. P., Kurapati Vishnu Sai Reddy, A. M. Kiruthik, Dr. J. Arun Nehru (2022) Prediction of Kidney Stones Using Machine Learning

[2] Türk C, Neisius A, Petrik A, Seitz C, Skolarikos A, Thomas K (2018) EAU Guidelines on urolithiasis. EAU Guidelines Ofce, Arnhem

[3] Pearle MS, Goldfarb DS, Assimos DG, Curhan G, Denu-Ciocca CJ, Matlaga BR, Monga M, Penniston KL, Preminger GM, Turk TM, White JR (2014) Medical management of kidney stones: AUA guideline. J Urol 192:316–324

[4] Dion M, Ankawi G, Chew B, Paterson R, Sultan N, Hoddinott P, Razvi H (2016) CUA guideline on the evaluation and medical management of the kidney stone patient—2016 update. Can Urol Assoc J 10:E347–e358.

[5] Taguchi K, Cho SY, Ng AC, Usawachintachit M, Tan YK, Deng YL, Shen CH, Gyawali P, Alenezi H, Basiri A, Bou S, Djojodemedjo T, Sarica K, Shi L, Singam P, Singh SK, Yasui T (2019) The Urological Association of Asia clinical guideline for urinary stone disease. Int J Urol 26(7):688–709

[6] Ganesan C, Thomas IC, Song S, Sun AJ, Sohlberg EM, Kurella Tamura M, Chertow GM, Liao JC, Conti S, Elliott CS, Leppert JT, Pao AC (2019) Prevalence of twenty-four hour urine testing in Veterans with urinary stone disease. PLoS ONE 14:e0220768

[7] Hsi RS, Sanford T, Goldfarb DS, Stoller ML (2017) The role of the 24-hour urine collection in the prevention of kidney stone recurrence. J Urol 197:1084–1089

[8] Goldfarb DS (2019) Empiric therapy for kidney stones. Urolithiasis 47:107–113

[9] NICE (2019) NICE Guideline—renal and ureteric stones: assessment and management. BJU Int 123:220–232

[10] Robertson WG (2006) Is prevention of stone recurrence fnancially worthwhile? Urol Res 34:157–161

[11] Z. Jadi, A Step-by-Step Application of PCA, Available on- https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[12] Available on- https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in neural information processing systems, vol. 30, 2017.