Introduction

Healthcare dataset is important to understand the differences and address the health imbalances in patient's population. Usually the healthcare dataset will consist of the information about the services that a particular patient is attaining during the experiences with the healthcare system. So, in this project we are using healthcare data in order to understand the machine learning algorithms and evaluate the algorithms.

For example, consider an GLU which means Glucose, if any patients the GLU values are high so we have to understand that the particular patient condition is worse. So, as we understand that we are concentrating on the values of dependent variables, the values may be continuous. Therefore, this type of tasks or problems will be a regression problem. By this project we will compute which algorithm produces the least errors.

Supervised machine learning algorithms:

- Supervised machine learning is that we train the machine using the data that is "labeled" nicely. And as it learned from labeled data it will help to predict the outcomes of unpredicted data.
- Supervised machine learning allows to collect data or produce the data output from the previous experiences. It also helps in solving real world computational problems. There are two types of techniques in supervised learning i.e. classification & regression.

Unsupervised machine learning algorithms:

- Unsupervised machine learning techniques is something that deals mainly with the "unlabeled data". This helps to find out unknown patterns of the data. And as we know it is easier to get the unlabeled data than the labeled data from the computer.

Metrics to evaluate machine learning algorithms:

- The metrics that are chosen for the evaluation of the machine learning algorithms are most important.
- The choice of the metrics that influence the performance of the machine learning algorithm are measured and compared.
- The metrics evaluation can be done for both classification and regression types of machine learning problems, but the project is a regression problem so we are concentrating on the regression metrics.
- A Regression metrics has most common metrics for evaluating prediction on the regression machine learning problems:
    1. Mean Absolute Error:
    2. Mean Squared Error:
    3. Root mean squared Error:
    4. Mean Absolute percentage Error (MAPE):
    5. Accuracy:

Mean absolute error (MAE):

- The difference between the actual values or true values and the values, which are predicated are said to be Mean absolute error.
- Mean absolute error (MAE) = True value – Predicted values.
- Hence, the mean absolute error takes the average of this error from every sample in a dataset and gives the output value.

Mean Squared Error (MSE):

- Mean squared error is calculated by taking the average of the square of the difference between the original and predicted values of the data.

$$\text{Mean Square error} = 1 / n \sum_{i=1}^{n} (Actual\ \ Value\ -\ Predicated\ values)^2$$

- The N is the total number of the observations in the dataset.
- In most of the regression problems, mean squared error is used to determine the performance of the model.

Root Mean square Error (RMSE):

- The Root Mean Square is equivalent to the Mean squared error but while determining the accuracy of the model the root of the value is used.

Mean Absolute percentage Error (MAPE):

- The mean absolute percentage error is nothing but the mean absolute of actual values - predicted values by predicted values.

Accuracy:

- 100 - Mean absolute percentage error.

Experimental setup:

1. **Data Description:**

The dataset that we have chosen is healthcare; it has 1930 instances, with 44 features.

- Information related to the data: Id --> Consider user id. Gender 1M --> Consider gender 1 as male. Eta --> Estimated time of arrival. sbp --> Systolic Blood Pressure. dbp --> diastolic blood pressure. GLU --> Glucose.

2. **Data processing:**

- Replace the NaN: If the dataset has NaN values, we are going to replace the NaN values with '0'. which is the most important step in the cleaning process. Though there are enough chances to replace NaN values with mean values. But I have chosen zero to replace the NaN values.
- Drop unnecessary columns: In my case, I have dropped the Data of birth because it is not so useful for the Prediction.
- Then, transform the features by scaling each feature to give range. This estimator scales and translates each feature individually such that it is in the given range on the training set, that is between the 0 and one using MinMax Scaler.

3. **Predict value:**

- In this project, we have changed the predicted values like: Hemoglobin, Apathy scale, Insulin, Dementia, Glu, Triglycerides del sangue (trigl_sangue) based on that we can understand which patient/person conditions are worst.

4. **Test Train split**

- For creating the machine learning model, it is important to evaluate how well it is able to map the inputs to output. In machine learning, first split the dataset or training data into a set for training and a set of testing. For this there are no specific rules as to exact size to make but it sensible to reserve a large sample for training. A typical split is 80% training and 20% testing data.
- The example image of after splitting the data with size of 24% testing data.
- Also, it is important in splitting the dataset, the splitting of the data should be randomly split so that you are able to get a good representation of the patterns that exist in the data in both training sets and testing sets.

```
# spliting the dataset into X_train, X_test, y_train, y_test:
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42)
# standarscalization
#standarscaler_X = StandardScaler()
```

Supervised:

- The regressions are choices as the only intention is understanding the different types of algorithms and by performing all possible regressions in this project, we can compute which algorithm is giving less error rate compared to others. After processing the baseline algorithm, I have tried to implement the grid search so that the error rate is varying to the baseline and able to obtain the best parameters of the algorithm.

Dummy regressor:

- Firstly, the train data is used in a dummy regressor to get a baseline score to further iterations of model development. For this we used Scikit-learn, a python library, which provides many supervised and unsupervised learning algorithms, by using the scikit-learn that is able to code in a single line.
- That allows training a model and making predictions based on simple rules, such as predicting at random.
- Then the evaluation is done by:

Mean Absolute Error	0.051278431372549005
Mean Absolute Percentage Error:	33.37603451549454
Mean Squared Error:	0.008047944852419244
Root Mean Squared Error:	0.08971033860386017

Decision Tree:

- The decision tree algorithm belongs to the supervised learning algorithm. The decision tree algorithm can be used for Classification and Regression problems too. Decision tree algorithms are a powerful prediction method and extremely popular. The goal of the decision tree algorithm is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules. The final model of the decision tree is so easy to understand by practitioners. The final decision tree can explain exactly why a specific prediction was made, making it very attractive for the user.
- In the Decision tree, for predicting a class or target variable for record, it starts from the root node / root of the tree. Compare the values of the root attributes with the records attribute.

Related Terminology for Decision tree:

- Root node: It represents a sample, and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing the node into two or more sub nodes.
- Decision node: When the sub node splits into further sub nodes, then we can say that as a decision node.

- Terminal node: Nodes that do not split into sub nodes are called Terminal nodes or Leaf nodes.
- Parent and child node: A node, which is divided into sub nodes, is called parent node and the sub node is called child node.

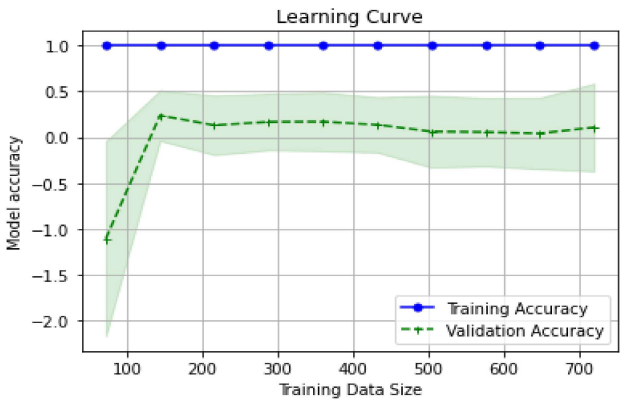
Regression Tree:

- A regression tree is used when the dependent variable is continuous. The value obtained by the leaf node or child node in the training data is the mean response of the observation falling in that region. It will only take discrete values in the test set, when the means of even if the dependent variable in the training data was continuous. A regression tree follows a top down greedy approach.

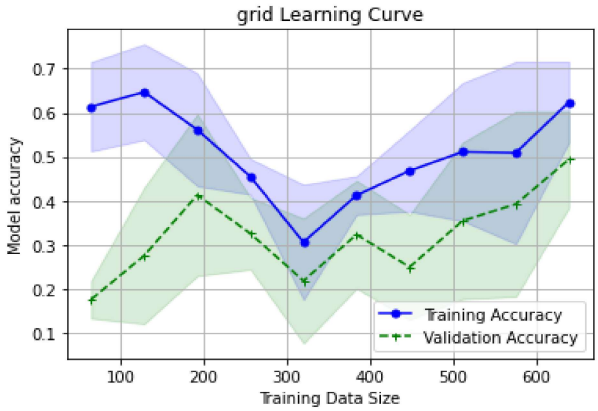
Evaluating of regression tree:

- We created an object of the class as Decision tree regression, which stores the address in the variable of the tree. Then fitting the tree with X Train and y train of the dataset. Then we predicted the values.
- The decision trees regression normally uses the mean squared error (MSE) to decide to split a node in two or more sub-nodes. For each subset, it will calculate the MSE separately. Then the tree chooses the value which results in the smallest MSE value.
- After implementing the grid search on decision tree 7the best parameters of the decision tree are:

{'max_depth': None, 'max_features': 0.7, 'min_samples_leaf': 1, 'min_samples_split': 0.5}



(fig 1: Decision tree regressor learning curve)



(fig2: After grid implementation learning curve)

	Decision tree regressor	Grid Search
Mean Absolute Error	0.04518372703412073	0.03576859195293035
Mean Squared Error	0.004406727702344293	0.00268721293435845
Root Mean Squared Error	0.06638318840146422	0.051838334602477824
Mean Absolute Percentage Error	26.34114875938484	25.54123094408792
Accuracy	73.65885124061516	74.45876905591209

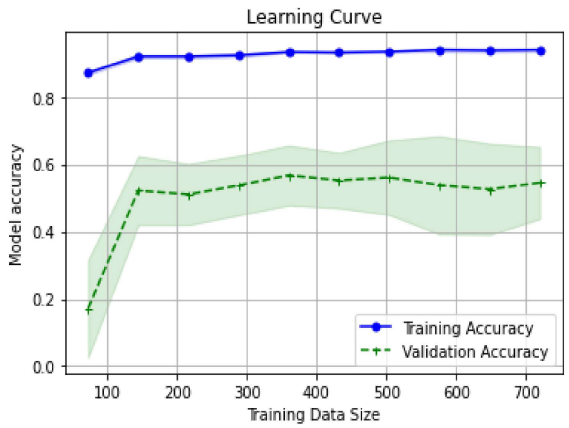
- We achieved a good improvement in accuracy of 1.09%, depending on the using grid search which are the most promising hyperparameters.

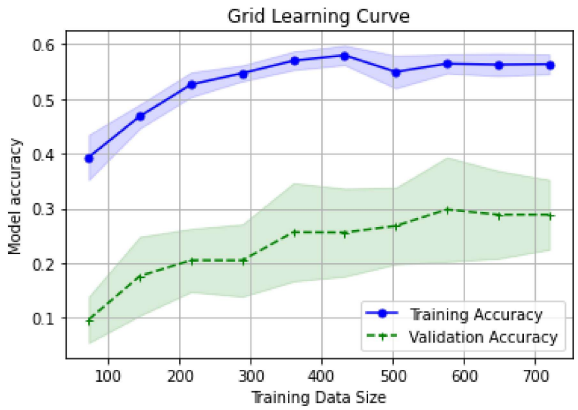
Random forest:

- Random forest is a type of supervised machine learning algorithm, and its algorithms are based on ensemble learning. The ensemble learning is a type of learning where two or more types of algorithms join different types of algorithms or same algorithms multiple times to form a more powerful prediction model.
- The random forest algorithms combine multiple algorithms of the same type that are multiple decision trees, resulting in a forest tree, then named “Random forest”. The random forest algorithm can be used for both regression and classification tasks. Effective ensemble algorithm as each decision tree fits on a slightly different training dataset, and in turn, has slightly different performance. Unlike normal decision tree models, such as classification and regression trees, the trees used in ensemble are unpruned, making them slightly overfit to the training dataset.
- Basic steps are involved in performing the random forest algorithm.
- Pick N random records from the dataset and build a decision tree based on the N records based on the number of trees in algorithms, it will repeat up to reaching max number of trees.

Evaluating of Random forest for regression:

- The random forest for regression class of the sklearn library is used to solve the regression problem for the random forest. The most important parameter of the random forest regressor class is the number of estimators of the parameters.
- The parameters are defined as the number of trees used in the random forest. For this accuracy random forest regressor we have used 20 to see how our algorithm performs.
- And also, the grid search best parameter of random forest is:
  {'bootstrap': True, 'max_depth': 80, 'max_features': 3, 'min_samples_leaf': 3, 'min_samples_split': 12, 'n_estimators': 100}





	Random Forest regressor	Grid Search
Mean Absolute Error	0.03304724409448819	0.04083405737440719
Mean Squared Error	0.002476373647191739	0.004641103324223339
Root Mean Squared Error	0.04976317561401944	0.06812564366098378
Mean Absolute Percentage Error	24.45463682694479	29.685244900042544
Accuracy	75.54536317305521	70.31475509995745

- We understand the improvement went negative compared to baseline which is -6.92%, which is a bad improvement.

Support vector machine.

- The support vector machine (SVM) is a supervised machine learning algorithm. The support vector machines are considered to be a classification approach, but it can work for both classification and regression type of problems. The support vector machines generate optimal hyperplanes in an iterative manner, which is used to minimize an error.
- The core idea of the support vector machine is to find a maximum marginal hyperplane which is MMH that best divides the dataset into classes.

The evaluation of support vector machine:

- First train the support vector regressor on the training data for that Scikit learn contains SVM(Support vector machine) library, which contains built-in classes for different SVM (support vector machine) algorithms because to perform a regression task, the support vector regressor class, which is written as SVR in the Scikit learn SVM library. This class takes one parameter, which is nothing but kernel type.
- And also, the grid search best parameter of Support vector regressor are:

  {'C': 1.5, 'epsilon': 0.1, 'gamma': 1e-07, 'kernel': 'linear'}

	Support vector regressor	grid Search
Mean Absolute Error	0.041944814534902335	0.04248526907407401
Mean Squared Error	0.00379621052488327	0.00387116820808992
Root Mean Squared Error	0.06161339566103519	0.06221871268428752
Mean Absolute Percentage Error	34.36615285253419	34.67111250999112
Accuracy	65.63384714746581	65.32888749000888

- We understand the improvement went negative compared to baseline which is -0.46%, which is a bad improvement of support vector regressor.

KNN:

- K-Nearest Neighbour (KNN) is the algorithm which can be used in both classification and regression problems. This uses *Feature similarity* to predict the values of any new data points.
- Initially it should be calculated the distance between the new point and each training point. And there are different methods to calculate this distance, from which the most known methods are *Euclidian, Manhattan & Hamming Distance.*
- Further to select the K Value, it has to be determined the number of neighbors we look at when we assign the value to any new observation.

The evaluation of KNN:

- From the range of 1 to 20, we have made the iteration of n_neighbors values, then fit the model and make predictions on the test set.
- After the grid search, then the best parameters:

  {'n_neighbors': 17}

	KNN	grid Search
Mean Absolute Error	0.047418635170603664	0.0472888683032268
Mean Squared Error	0.006704553220217551	0.00676290702062373
Root Mean Squared Error	0.08188133621416759	0.08223689573800637
Mean Absolute Percentage Error	32.64655968220745	32.65076442327052
Accuracy	67.35344031779255	67.34923557672948

- After applying the grid search the accuracy has decreased by almost 0.01%, so we can consider the KNN with the baseline accuracy and grid search accuracy to be almost the same.

AdaBoost:

- Boosting refers to a class of machine learning ensemble algorithms where models are added sequentially and later models in the sequence correct the predictions made by earlier models in the sequence.
- AdaBoost, short for "*Adaptive Boosting*," is a boosting ensemble machine learning algorithm, and was one of the first successful boosting approaches.

An AdaBoost regressor:

- The AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases.

- The best parameters are:

  {'base_estimator__max_depth': 2, 'n_estimators': 10}

	Adaboost	Grid Search
Mean Absolute Error	0.03735392790116358	0.03198991303345863
Mean Squared Error	0.002721545179930666	0.0023291246399552894
Root Mean Squared Error	0.05216843087472218	0.048261005376548975
Mean Absolute Percentage Error	31.426781315615205	24.18168285958855
Accuracy	68.5732186843848	75.81831714041145

- The Improvement of Adaboost is 10.57%, which is very good. If we compare both the adaboost with grid search, the best parameter is accuracy is good.

Neural Networks:

- Neural network is the machine learning framework that attempts to mimic the learning pattern of natural biological neural networks, and also neural networks are used to solve the challenging artificial intelligence problems. The biological neural networks have interconnected neurons that receive the inputs and based on the inputs they produce the output signal through an axon to another neuron. The main advantages of neural networks are non-linearity, variable interactions and customizability.
- The process of creating the neural network begins with the perceptron. In simple terms, the perceptron receives the inputs then multiplies the input by some weights and passes them into an activation process such as logistic, relu, tanh and identity to produce an output. By adding the layers of these perceptrons together, known as a multi-layer perceptron model. The layers of the neural network are three, they are input layer, hidden layer and output layer.
- The input layer directly receives the data.
- The output layer creates the required output, and in between the input and output layers are known as hidden layers where the intermediate computation takes place.
- A neural network algorithm can be used for both classification and regression problems.

The evaluation of Neural network:

- Created a neural network model with multilayer perceptron regressor, the model with the hidden layer size three layers, which has the same number of neurons as the count of features in the dataset. And also selected ‘Relu’ as activation function, means a rectified linear unit and it is not linear and has the same benefits as sigmoid but with better performance and ‘Adam’ as solver for the weight optimization means an adaptive learning rate optimization algorithm that is designed specifically for training neural networks.

- Multilayer perceptron regressor fits the model to the training data and trained model to generate the prediction of the training and test dataset respectively.

	Neural network
Mean Absolute Error	0.0563220842608471
Mean Squared Error	0.0068608155100252895
Root Mean Squared Error	0.0828300399011451
Mean Absolute Percentage Error	43.45893312814744
Accuracy	56.54106687185256

overall:

	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	Accuracy	Grid Search
Decision tree	0.045	0.004	0.066	73.658	74.4587
Random Forest	0.033	0.002	0.049	75.545	70.314
SVR	0.041	0.003	0.061	65.633	65.328
KNN	0.047	0.006	0.081	67.353	67.349
AdaBoost	0.037	0.002	0.052	68.573	**75.818**
Neural Network	0.056	0.006	0.082	56.541	

- By the above table, we can observe the baseline accuracy of Random forest is 75.545 and grid search of the AdaBoost's accuracy is 75.818, which are high compared to others. By this I can conclude that these both have good accuracy and less error rate so they are the best models for this project. Now considering only these techniques I have tried to implement other features like I mentioned above in the predict value.

Features are predicted	**Accuracy**	
	Random forest	**AdaBoost Grid search**
GLU	75.545	75.818
Hemoglobin	86.215	84.413
TRIGL_sangue	48.708	43.346

- By above table, we can observe the features that I have predicted i.e; GLU, Hemoglobin and TRIGL_sangue when compared all the accuracy of random forest is better than AdaBoost grid Search.

Unsupervised:

- When it comes to unsupervised learning, the clustering is an important. Because it is mainly dealing with finding a structure or pattern in a collection of uncategorized data. Unsupervised Learning Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data.
- By this in this project we are going to see the patient's condition by consider an GLU which means Glucose, if any patients the GLU values are high then that patient fall into the cluster so we have to understand that the particular patient condition is worse. Likewise, another features we have used (which we have used in supervised learning).

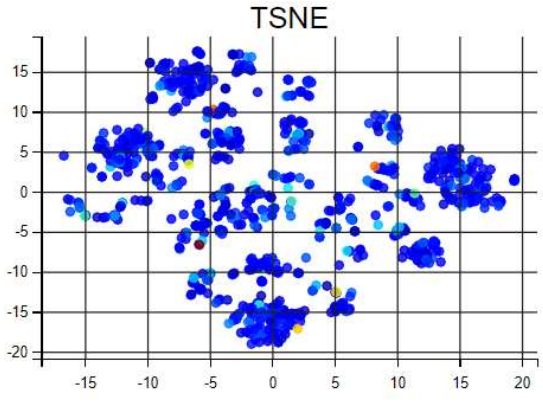
Dimensionality reduction

t-SNE:

- t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and **visualizing high-dimensional data. In simpler terms**, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space.
- The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.

Implementation:

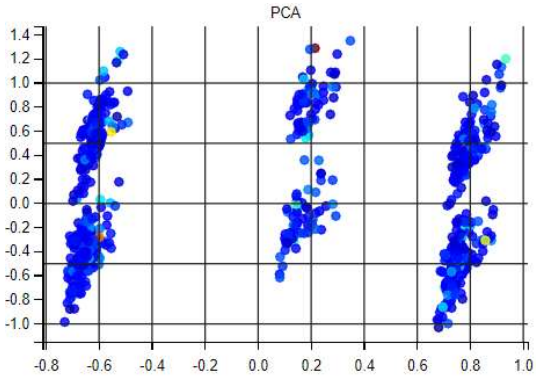
- The parameters are used in the t-SNE are n_components which is dimension of the embedded space, verbosity level. The perplexity, which is related to the number of nearest neighbors that is used in other manifold leaning algorithms.



PCA:

- Principal Component Analysis or PCA is a linear feature extraction technique. It performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. It does so by calculating the eigenvectors from the covariance matrix. The eigenvectors that correspond to the largest eigenvalues (the principal components) are used to reconstruct a significant fraction of the variance of the original data.

- In simpler terms, PCA combines your input features in a specific way that you can drop the least important feature while still retaining the most valuable parts of all of the features. As an added benefit, each of the new features or components created after PCA are all independent of one another.

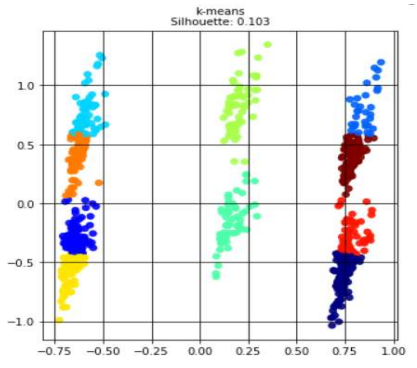


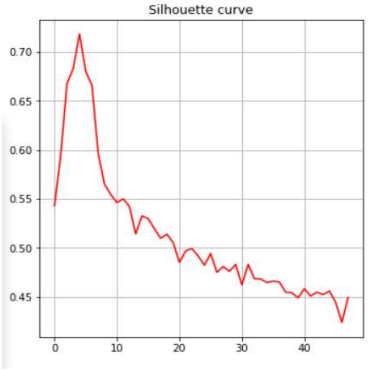
Clustering on already reduced training set:

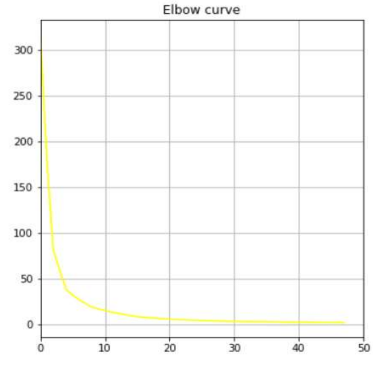
K-means:

- K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.
- The output of the algorithm is a group of "labels." It assigns data point to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.
- Generally, the silhouette score falls within the range [-1, 1].
- The silhouette score of 1 means that the clusters are very dense and nicely separated. The score of 0 means that clusters are overlapping. The score of less than 0 means that data belonging to clusters may be wrong/incorrect.
- The silhouette plots can be used to select the most optimal value of the K (no. of cluster) in K-means clustering.

K-PCA:



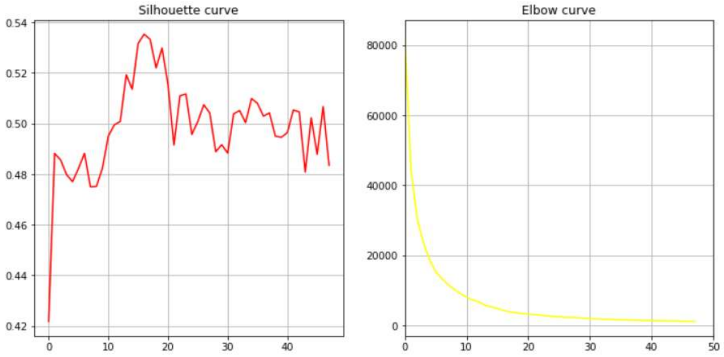


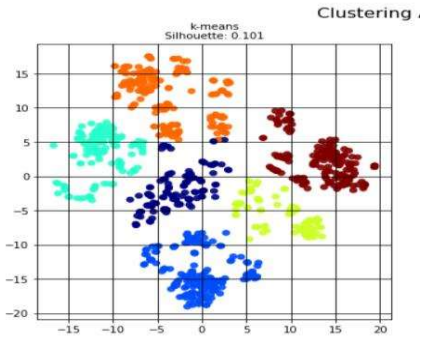


- The test was carried out with the both the silhouette and elbow curve, with the k = 6. Which means the clusters has highest score of silhouettes is 0.7181705622563843 and lowest score of silhouettes is 0.4350978029116942.

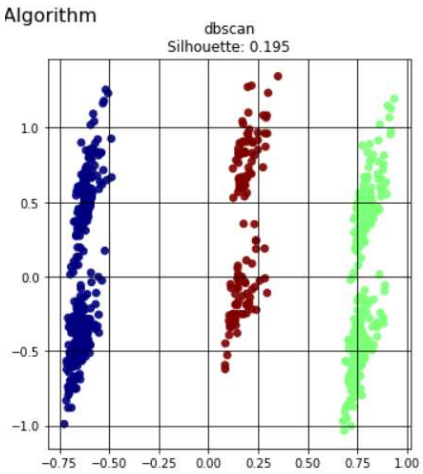
- The silhouette score which are near to 1, so we can conclude that the clusters are very dense and nicely separated.

K t-SNE:





- The test was carried out with the both the silhouette and elbow curve, with the k = 6. Which means the clusters highest score of silhouettes is 0.5353059768676758 lowest score of silhouettes is 0.4218114912509918.
- And the silhouette score of k_means with prediction values are far away from the PCA, so we can conclude that the clusters are not good which compare to the PCA.



DBSCAN

- Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm, Perform DBSCAN clustering from vector array or distance matrix.
- DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density.
- By above the k means we have understood that the PCA dimension reduction is best, so for the DBSCAN we are using the only the PCA dimension reduction and finds the clusters.
- After applying the PCA on DBSCAN the silhouette is 0.195.

Conclusion:

- From the all above study and work, I have understood that how to work with machine learning algorithms and according to supervised learning algorithm “the random forest is having best accuracy and less error rate” for predicting the features of the dataset. And I want to conclude that by this project we can understand which algorithm is best for this project, and with 86.21% accuracy random forest is the best, that we will be able to predict the hemoglobin, and according to unsupervised the k-means with PCA dimension reduction have the silhouette value which is good, so that the clusters are very dense and nicely separated.

References:

1. Lin Gao, Feng Gao, Xiaohong Guan, Dianmin Zhou and Jie Li, "A Regression Algorithm Based on AdaBoost," 2006 6th World Congress on Intelligent Control and Automation, 2006, pp. 4400-4404, doi: 10.1109/WCICA.2006.1713209.
2. Ajay Ohri, [2017 February 16], 8 Popular Regression Algorithms In Machine Learning Of 2021, https://www.jigsawacademy.com/popular-regression-algorithms-ml/
3. SUNIL RAY, [2017 SEPTEMBER 9], Commonly used Machine Learning Algorithms, https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
4. Ajitesh Kumar, [2020 September 17], KMeans Silhouette Score Explained With Python Example, https://dzone.com/articles/kmeans-silhouette-score-explained-with-python-exam
5. Jason Brownlee, [2016 May 25], Metrics To Evaluate Machine Learning Algorithms in Python, https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/
6. Manish Pathak, [2018 September 13], Introduction to t-SNE, https://www.datacamp.com/community/tutorials/introduction-t-sne