# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 3

Name: Ravi Varma Kumar Bevara

### **PART I: Model and Dataset Description**

Unsupervised learning

It deals with problems in which data doesn't have labels. That property makes it very problematic for many applications as absence of labels representing the desired behavior for your model means the absence of solid reference point to judge the quality of model.

Clustering - It is a problem of learning to assign a label to examples by leveraging an unlabeled dataset. Because the dataset is completely unlabeled, deciding on whether the learned model is optimal is much more complicated than in supervised learning. K-means clustering algorithm works in the following steps...

* We will choose an arbitrary k-value which represents the number of clusters.
* Then, we randomly put k feature vectors, called "centroids".
* Now, we will compute Euclidean distance from each point to each centroid and assign the closest centroid to each example in training data.
* Then, we calculate the average feature vectors to each set centroid, as they will become new locations of centroids.
* We recompute the distances from each example to each centroid, update the centroids until the assignments don't change even after the centroids are recomputed.

This data set is a clinical trial data set which is collected from more than 10,000 patient for different assessments called for patients participated in industry, foundation, and sponsored trials. The actual data set is collected regarding with the motivation in observing the trend of the ALS disease after the diagnosis for different demographic patients over time.

In total, the training data set have 2223 observations with 101 attributes and the test data is comprised of 131 attributes with 78 observations. Original data set how many revised versions including demographic data, family and medical history data, adverse events history data, symptoms send outcome measures data etc. The version of the data set chosen for this assignment is regarding answering the question “What patient phenotypes are automatically and reliably detected to predict the slope or functional score of the disease over time after diagnosis”.

The data set includes blood interiors such as proteins which includes albumin, electrolytes including calcium, potassium, sodium, bicarbonate, chloride, magnesium, kidney test, liver test, blood cell count at aggregated levels including minimum value, maximum value, median value, and range for each of the phenotype of a patient.

The bigger picture in this assignment is predict the outpatient phenotypes can be automatically and reliable determined along with predicting the change of ALSFRS slope change using the holistic patient specific data and also protecting the survival of patients at a given time point post diagnosis.

### **PART II: Exploratory Data Analysis (Data Pre-Processing, Data Visualizations)**

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

A picture containing application

Description automatically generated

To better understand the variables, we need to look how the data is distributed and what’s the shape of the data. I continued with exploring the columns of the data. In this process, I recognized values of the target variable ‘quality’ are not balanced or ordered. The modules NumPy and pandas has various functions such as .head() to check the structure of the data. Also .columns will result all the columns of the dataset.



Graphical user interface, table

Description automatically generated

After looking at the bird’s eye view of training and testing data, I decided to observe an exploring statistical measure such as standard deviation quartile distribution for each variable inside training and testing data.

Table

Description automatically generated

Based on the above screenshot, we can clearly see that the given raw form of data is too much variant for each variable. So, we may need to use the scaling property to get all the data points into a same range, which will eventually gain an advantage in model training and testing.

Also from the below image, we can see that the test data have 30 more extra vegetables compared to the top trend data, so we may need to consider that opinion in model training and testing phase.

Graphical user interface

Description automatically generated with medium confidence

Also, we can see that there are no null values in any column, from the response in the above picture. Once, I came to know about the variables, then I wanted to get a better understanding of each variable in the dataset separately. I think the correlation plot would be a better choice to examine the relationships between target and other variables, along with histograms of each variable.

Graphical user interface, application

Description automatically generated

After observing the training data and testing data, I found that two particular attributes such as ID and subject ID are not showing any importance in this particular analysis because they are not phenotypes of patients as ID is representing the unique number for the clinical trial and subject ID is representing the unique ID for each patient in that particular clinical trial. So, from the above information, I decided to drop both of the variables from training and testing data.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

When variables used in clustering collinear, some variables get a higher rate than others. If the wearables are perfectly correlated, they effectively represent the same concept. Does even though the clustering analysis deals with the numerical coordinates, the correlations between the variables have an effect on the results of the analysis. So, I created the correlation matrix using heat map visualization…

A picture containing text

Description automatically generated

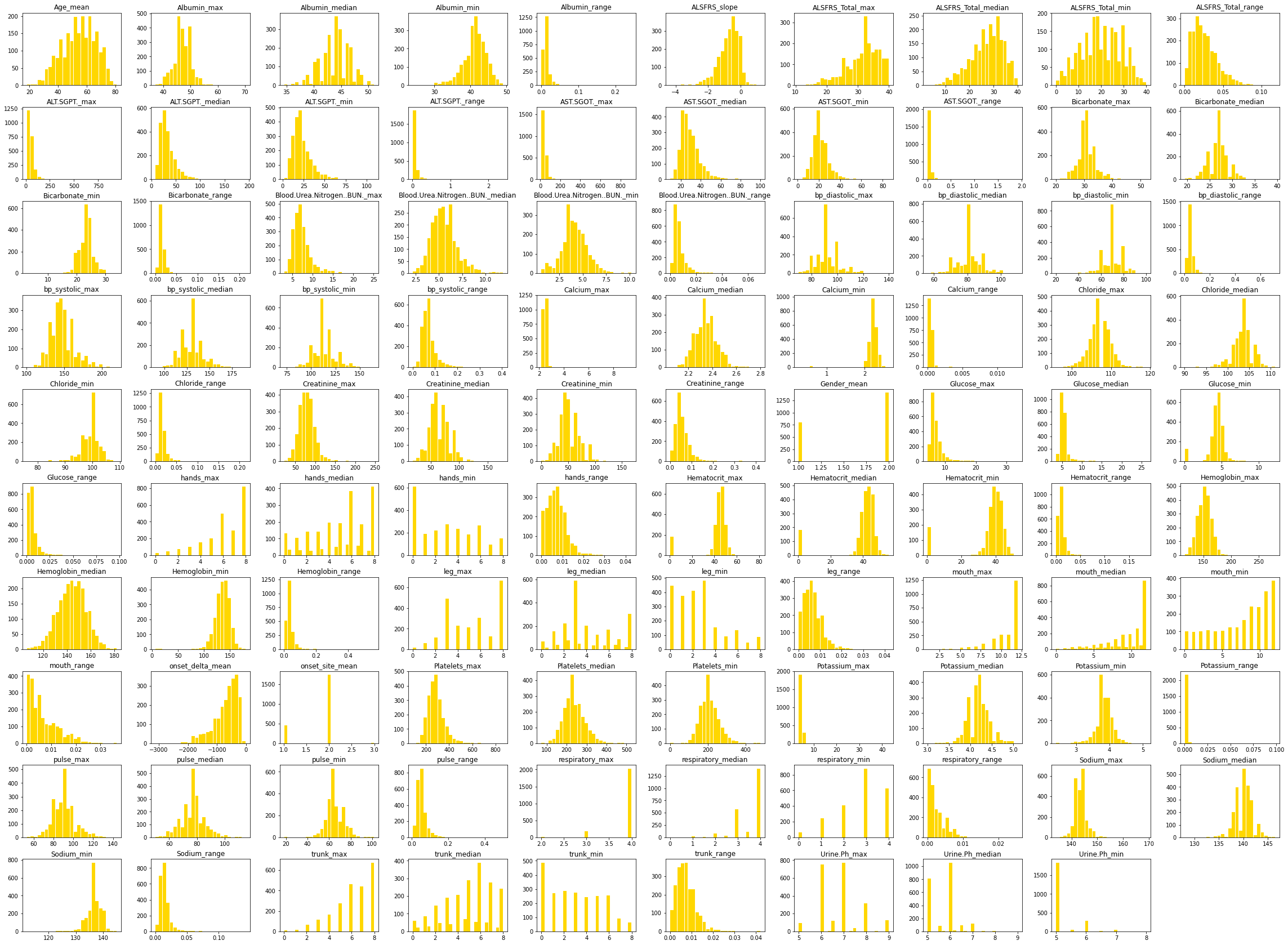
Chart, scatter chart

Description automatically generated

After looking into the correlation matrix result, we can list the variables with strong positive correlation, moderate positive correlation, and low positive correlation. Also, the same cases with the negative correlations. But due to the unclear image of the correlation matrix, I decided not to deal with the correlations as it is very complex with 101 variables in the data set both in training and testing.

Even though I didn’t consider the correlation between the variables, I decided to observe and explore the distributions of each variable in training data to observe this skewness and distributions and deal with the skewness.





After observing the histogram plot for each distribution in the variables, I felt my binning was default to some particular attributes, then I decided to look the density of each variable which will give me more clear view for the distributions of the attributes.

Graphical user interface, text

Description automatically generated

A picture containing shoji, building

Description automatically generated

After checking the distributions of all the variables, I opted not to transform any variable with a log or square root transformations as we are not dealing with any regression or classification problem. As we are dealing with clustering problem, which is unsupervised learning, we may need to scale and standardize the units of all the variables, as scaling represents to change all the numbers into the same range, I have chosen to normalize the variables to be more precise and clearer to the machine.

To start off with, I plotted the optimal key value by taking 15 iterations using the elbow method, I can see that minimum value of k to be chosen is 2 as we can see the error rate, which is “Within Cluster Sum of Square Errors” has a steep decrease, so I decided to choose the initial k value to be 2.

Graphical user interface, text, application, email

Description automatically generated

Shape

Description automatically generated

After choosing the initial K value, I decided to reduce the number of features by checking the feature importance in the data. As the research problem is dealing with ALSFRS slope, I separated that column from the data set and trained a random forest model to plot the features importance. But after looking at the results, we can see that ALSFRS total range, which is also a metric such as low in defining the functional rate of the disease after the diagnosis is strongly affecting the target variable. The next nine features are almost at the same scores of feature importance, I decided not to decrease the number of features with using the feature importance. Also, as we are dealing with the problem in a healthcare, we need to consider the importance of each attribute in its own field, suppose the age might not be a good factor to be considered but it still makes sense when we are finally answering the research problem which is the disease rate after the diagnosis.

Graphical user interface, text, application, email

Description automatically generated

**Timeline

Description automatically generated with low confidence**

As discussed earlier, I will be using the standard scaler from Scikit learn module to standardize the data and also normalize the data upon standardization.

Graphical user interface, text, application, email

Description automatically generated

After standardizing the data, to reduce the number of dimensions that the machine has to look decided to use PCK which is principal component analysis, indeed forming eigenvectors representing the 101 features in their own orthogonal spaces.

Graphical user interface, text, application, email

Description automatically generated

A picture containing shape

Description automatically generated

From the above image, we can see that 20 components are almost explaining the 95% variance in the data, so I decided to fit the principal components with 20 as number of components. Then I use the same standardization, normalization and calculated the principal components with the same number as I need to balance the train and test data which same number of dimensions.

Graphical user interface, text, application, email

Description automatically generated

### **PART III: Model Training (Splitting the Data, Applying the Model)**

After a deep study in the packages available for model training in python, I decide to use sklearn also known as Scikit-Learn to train my k-means algorithms. After looking at the data in the preprocessing phase, I have decided to build 6 models and compare the best of the models by using different metrics and selection methods. To start off with, I will be choosing k value to be 2, respectively for training and testing phases of the model.

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

As we can see the initial K value is differentiating very good clusters in the total dataset, now we will be training and testing the dataset using rest of the k-values.

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

**PART IV: Model Evaluation Metrics (Prediction Results, Test Scores, and Metrics)**

Unlike to supervised learning where we have the ground truth to evaluate the model’s performance, clustering analysis doesn’t have a solid evaluation metric that we can use to evaluate the outcome of different clustering algorithms or models. Moreover, since key means requires key is an input and doesn’t learn it from the data there is no correct answer in terms of the number of clusters that we should have in any research problem or any industrial problem. Sometimes the domain knowledge which is healthcare in this case may help the intuition that the scientist or analyst have, but that usually does not help in all cases. In cluster predict methodology, we can evaluate how well the models are performing based on different key clusters since the groups are used in the top-down approach in decision making.

We will be performing some model evaluation metrics on all the above six models, to judge which is the best model based on performance of the model. The two metrics that I have chosen to decide on the best model are…

1. Elbow Method
2. Silhouette analysis

While the elbow meter gives us an idea on what is the code K number of clusters that are best to start the model training, it will be using the sum of squared distance between the data points and their assigned clusters centroids. We will pick K at the spot where the error rate which is squared distance summation within the clusters starts to flatten now forming an elbow. As we have already performed the elbow method before starting to train the model, we will use that method to compare the other metrics whether we are using good number of K in clustering.

Graphical user interface, text, application, email

Description automatically generated

Shape

Description automatically generated

From the above graph, I can see that K equal to two is the better choice compared to all other key values. Sometimes it is still hard to figure out whether the initial K value is a good number of clusters as the curve is monotonically decreasing and may not show any elbow or it doesn’t have any points with the curve starts to be flattening out.

The second metric which is Silhouette Analysis, can be used to determine the degree of separation between the clusters. The coefficients of these metric scores can take values in the interval training from [-1,1], where and which 0 represents the sample is very close to the neighbor in clusters, 1 represent the sample which is far away from the neighboring clusters and -1 doesn’t the sample assign two wrong clusters. So, we want to the coefficients to be as big as possible and close to 1 to decide on the good clusters.

Table

Description automatically generated

From the above picture, we can see that number of clusters being 2 is the best average silhouette score around 0.46. But generally speaking, good number of clusters will have available 0.5 silhouette average score as well as the number of clusters have higher than the average score. I tested both on train data and test data, to look out for the overfitting problem which is most common in machine learning. As we can see the same number of clusters being 2 in the test data he’s giving me the best score, but at a very low level 0.09.

So, I decided to visualize the error rates or discourse block for different number of clusters, in which I used train data to train me model and the test data to calculate the score.

Graphical user interface, text, application

Description automatically generated

Chart, line chart

Description automatically generated

As we can see from the above picture, number of clusters being 12 is the best choice in deciding number of clusters as it is the point where the error rate is negligible. I plotted the scattered data for train and test using these 12 numbers of clusters.

Graphical user interface, text, application, email

Description automatically generated

Chart, scatter chart

Description automatically generated

### **PART V: Summary and Conclusion**

In the conclusion, KMeans algorithm is good in capturing the structure of the data is clusters has a spherical like shape. It always tries to construct a nice spherical shape around the centroid, which is not a very good choice for clusters with complicated geometrical shapes. The main drawback of this algorithm is it doesn’t lead data points that are far away from each other share the same cluster even though they obviously belong to the same cluster. This can be overcome by single linkage hierarchical clustering method. Next, we will generate the data from different dimensions with normal distributions with its own means and standard deviation so our data is a group of multivariate normal distribution with a different mean I am single standard deviation is the scenario where this algorithm couldn’t figure out the clusters correctly as it will always try to minimize the Witten cluster variation, it gives more weight to bigger clusters than the smaller ones. In other words, the data that should belong to a small cluster maybe not considered informing the centroid as the algorithm concentrates on the larger cluster. And the different complex geometrical shapes in the data can be resolved by spectral clustering method.

Although K means clustering is one of the most popular clustering algorithms and usually the first thing that which will apply when solving an unsupervised task to get a better idea about the data, the goal in this problem is made somewhat clear regarding the patient phenotypes used to predict the death rate after the diagnosis for the disease. However, as We can see from the results, the data which is provided to us is cluster best into 12 groups. So, in my opinion we can have 12 groups of patient phenotypes Will be best reliable in predicting the change of ALSFRS slope over the duration after the diagnosis of the disease.