EP4130: Final Project

Devulapalli Sai Prachodhan
EE20BTECH11013

Nandyala Vishwaram Reddy
EE20BTECH11059

Indian Institute of Technology, Hyderabad — April 24, 2023

Abstract

This project has mainly four objectives where first objective is to compare existing and new medication, second being clustering patients based on initial measurement, predicting 18th week PANSS scores as third and at last addressing issue of classifying patient whether he passed test or failed or to be checked with clinical specialist depending on given data. This report will discuss results obtained after applying the data science and statistical methods on a medical dataset which tracks patients who are diagnosed with schizophrenia over time. The dataset contains thirty scores defining the severity of the symptoms of schizophrenia on the scale of 1-7 via the Positive and Negative Syndrome Scale (PANSS) along with country and visited days. Each time a patient is assessed, factors like the ID of the patient, the ID of the evaluator, the location and day of the assessment, and PANSS scores are recorded.

Keywords - Kernel Density Estimation, P-value Test, Correlation, Heatmaps, Null and Alternate Hypotheses, Freedman rule, LOWESS, GradientBoostingClassifier, GridSearchCV, FeatureExtraction, AIC, BIC, Gaussian Mixture modeling, KMeans, Linear Regression, Double Exponential Smoothing.

Introduction

As part of the project, we are working with collected data from five randomized controlled trials for patients with schizophrenia. Initially patients are run through a screening test whether they meet the criteria for further analysis and then anonymized drug is being evaluated over all five trials for its efficacy in treating schizophrenia. Patients in the trials are followed for varying amounts of time (depending on the criteria of the study) and observed for symptoms related to schizophrenia. At start, patients baseline measurement (i.e. measurement on visit day 0) is taken. At this time, they are randomized into one of the two treatment groups : treatment or control group. The control group is provided the "accepted" standard medication for treating schizophrenia, while the treatment group is provided anonymized medication. Throughout the study, the patient comes back for follow-up visits to have the same measurements repeatedly taken.

You can find all datasets from here and clear full problem statement here

0.1 PANSS Score

Positive and Negative Syndrome Scale (PANSS) is known as the "golden standard" that all assessments of antipsychotic behavioral disorders should follow. The PANSS has three classes of items that are assessed: Positive symptoms (7 items) which refer to an excess or distortion of normal functions (e.g. hallucinations and delusions), Negative symptoms (7 items) which represent a diminution or loss of normal functions, and General Psychopathology symptoms (16 items). Each PANSS item is rated on an ordinal scale from 1 (i.e. absent) to 7 (i.e. extreme).

0.2 Data Briefing

We were given 5 csv files named Study_A,Study_B,Study_C,Study_D,Study_E. After clear examination of data visually we had Study_A,Study_B,Study_C,Study_D having same fields whereas Study_E is lacking LeadStatus field which we have to predict in the 4^{th} objective. The below explains the fields of the each csv file (an image extracted from problem statement):

1. *Study* - A character indicating which of the five studies the data represents.
2. *Country* - The country where the assessment was conducted.
3. *PatientID* - An identification number given to each unique patient.
4. *SiteID* - An identification number given to each unique assessment site.
5. *RaterID* - An identification number given to each unique rater.
6. *AssessmentiD* - An idenfication number given to each unique assessment conducted.
7. *TxGroup* - A string corresponding to the patient's (randomly) assigned treatment group.
8. *VisitDay* - An integer corresponding to the number of days that have passed since the baseline assessment.
9. *P1-P7* - The scores corresponding to each of the 7 positive symptoms of the assessment.
10. *N1-N7* - The scores corresponding to each of the 7 negative symptoms of the assessment.
11. *G1-G16* - The scores corresponding to each of the 16 general psychopathology symptoms of the assessment.
12. *PANSS_Total* - The sum of of the ratings across the 30 PANSS items.
13. *LeadStatus* - A string indicating whether the assessment's audit passed, was flagged, or was assigned to a CS (i.e. clinical specialist).

1 Treatment Effect

Question 1

Does the (anonymized) treatment have an effect on schizophrenia?

1.1 Introduction

As said in the earlier, we have two groups namely Control and treatment group. Patients are grouped into these two groups to differentiate and interpret about effectiveness of the newly proposed medication to treat Schizophrenia. Patients grouped as Control were given standard or exisiting medication and those who were interpreted as Treatment were given ananomyzed or newly proposed treatment.Now the below is the methodology we have used to tackle this question:

1.2 Need of Checking initial distributions

We are assuming that we have considered their is no bias between the initial measures of Schizophrenia of patients. In reality, we might assign the worst hit patients to control group (standard medication) and patients with meagre symptoms to treatment group(treatment that is being evaluated) and vice-versa. This is data-snooping bias as we have already run our experiment and we are being asked post-hoc to conduct the statistical analyses which we in general get biased to new treatment results.

So, first we need to consider the distributions of the baseline measurements and verify they almost match. If the initial measurements match almost then we can compare further and if it is not the case we can't further compare.

1.3 Pre-Analysis

As said in the previous section, to plot the distributions of treatment and control groups, I have considered histograms of the given data using **freedman** rule to calculate optimal bin widths for histograms. Also, I have also used **Kernel Density Estimation (KDE)** to plot Kernel density plots using Gaussian Kernel.

I haven't plotted all plots because as we know from the above features description raterID, Study, Country, AssessmentID, PatientID, LeadStatus, VisitDay (always =0 here) had nothing to do with comparisions of initial distributions. I have also considered G_total, P_total, N_total abstracts all remaining individual fields like P_k, G_j, N_i etc. Using the code given ***here***, these are generated plots:

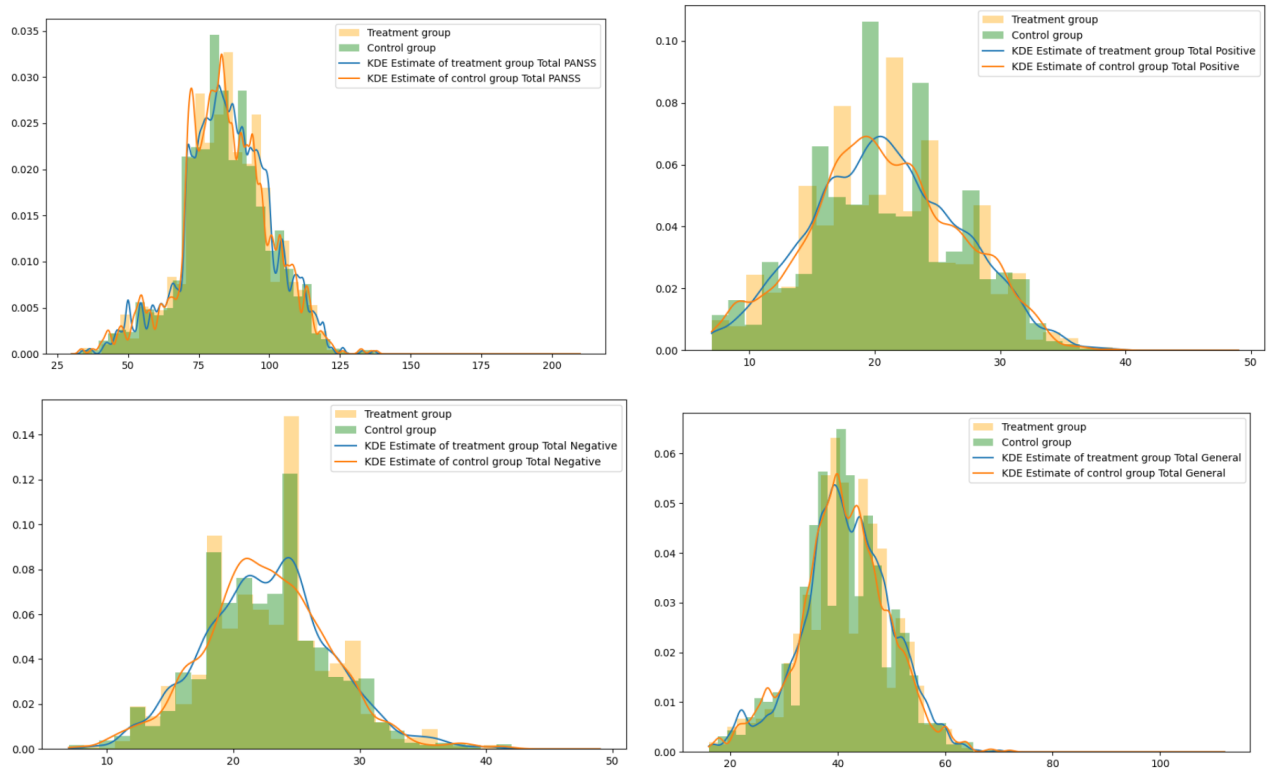


Fig - 1: Initial Distributions histograms as well as KDE plots for different features (a) Total PANSS Score (b) Total positive scale score (c) Total negative scale score (d) Total general scale score

The above are the plots for distributions of total PANSS_score, P_total, N_total, G_total across all studies combined for both treatment and control groups. From the above, it is clear that the histograms and KDE plots of baseline measurement distributions for both groups are almost colliding. This means we are starting with data that is almost unbiased for both the groups treatment and control. As the initial data is unbiased we can compare the data over visit days for both treatment and control groups.

1.4 Analysis

As we can compare two groups unbiasedly, we can apply data-science methods to verify effectiveness of newly proposed treatment (for treatment group) over standard medication (control group). Our concerned measure of treatment effectiveness is decrease in PANSS_score, P_total, N_total, G_total because as they decrease we can say that symptoms are approaching normality and thus treatment would be working effectively.

1.4.1 Using Correlation

I have considered plotting heat-map which shows correlation of one feature over other. Again, I have ignore fields like ID's which are some unique numbers generated but not used in analysis. The below correlation heat map shows the correlation between PANSS_score, P_total, N_total, G_total vs TxGroup is almost close to 0 over all studies.

To say about statistical significance of the above fact we have considered p-value statistic for correlation between measures vs TxGroup. The p-value for pearson correlation of PANSS_score vs TxGroup is 0.987(rounded to three digits). Our null hypothesis is that there isn't correlation between PANSS_Total and TxGroup. Consider alternate hypothesis that there is correlation between PANSS_Total and TxGroup. Then corresponding p-value, taking alternate hypothesis as null hypothesis, will be 1-0.987 which is lesser than 0.05. So we can reject the alternate hypothesis that there is correlation between PANSS_Total and TxGroup.

Note : In the above, we have used the fact that, p-value of null hypothesis + p-value of alternate hypothesis = 1 **if and only if** alternate hypothesis is conjugate hypothesis of null hypothesis. In the above case as we are considering alternate hypothesis as any non-zero correlation case which is exactly conjugate for no correlation (null hypothesis).

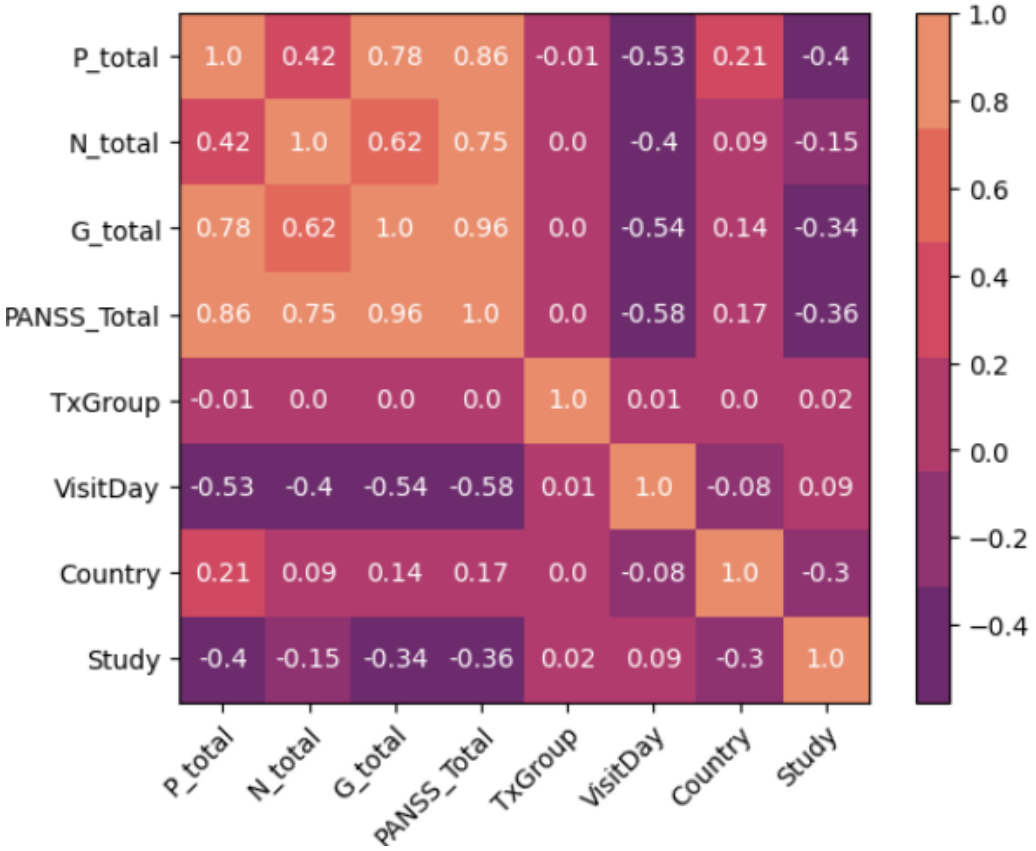


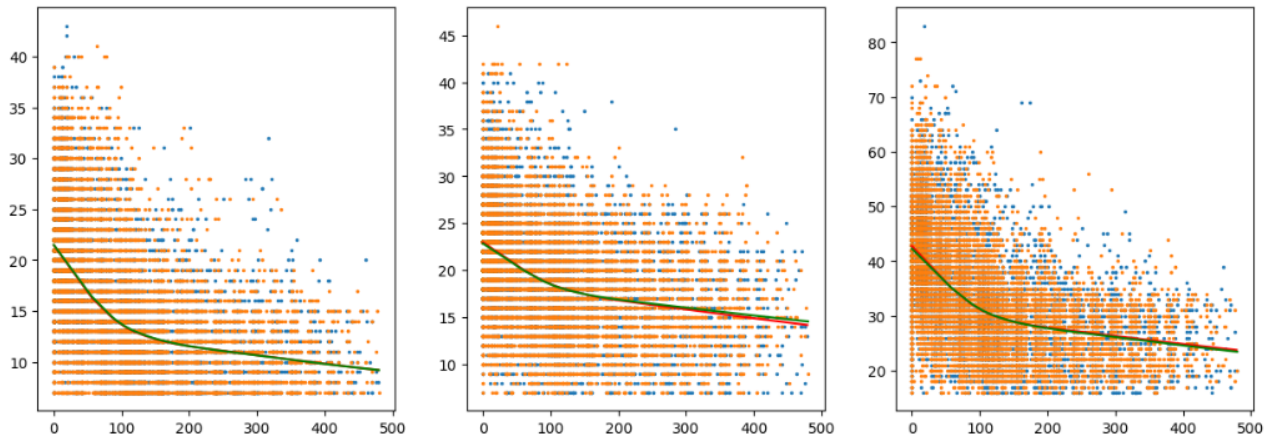
Fig -2 : The correlation map of one feature over other.

From the above we conclude that measures over all visit days is independent of two groups i.e there is **no improvement using new treatment** over standard medication. But doing the same thing for individual positive scale, negative scale, general scale gave p-value for null hypothesis as 0.173,0.482,0.584 that means we cannot reject null and also alternate hypothesis, concluding that the newly proposed treatment might be working on specific set of positive or negative or general class of symptoms well or vice-versa.

1.4.2 Using LOWESS and Scatter plots

Another way of tackling the above issue is by plotting measures against visitDay for both the control and treatment groups and also analyzing using locally weighted scatterplot smoothing(**LOWESS**) plots.

LOWESS is a non-parametric method which uses locally weighted linear regression (LWR) algorithm to fit smooth curve data depending on local weights. More info about it can be found here. For this project, we are using LOWESS function that is already implemented in "statsmodels" package.



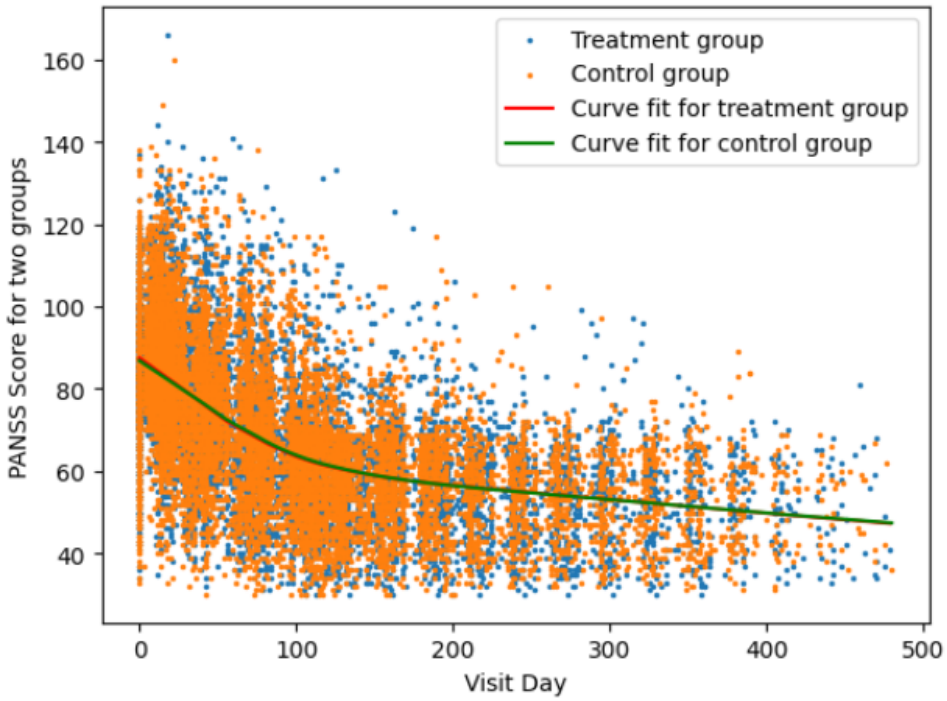


Fig - 3: Scatter and LOWESS plots (a) Total positive, negative and general scales score (from left to right). Red line denotes LOWESS plot for treatment group & green for control group. (b) Total PANSS Score

For treatment to be effective, the measures should decrease down quickly over the visit days i.e, curves which stay lower have better effect of treatment. From the above plots, we can clearly see that scatter points and LOWESS curves are also almost same (one on another). We can see noticeable deflection of the curves in Total negative and general scales only. From the plots, we can say that in overall sense the two groups are similar but there is slight advantage of standard medication over newly proposed in general class of symptoms concerned and newly proposed over standard in negative scale.

1.4.3 Curve fitting

In this we will analyse which data fits better to a given model. The model I considered is,

Note : I ignored "study" field from dataset, as I have to concentrate more on TxGroup and visit Day effect. I considered approximation that rater is unbiased and country has nothing to do with skewness of data.

$$\text{Score} = a + b * \text{visitDay} + c * \text{visitDay} * \text{TxGroup} \tag{1}$$

If c $\rightarrow$ zero, we can conclude that there is no improvement in new method. The below are results obtained:

Score	a (p-value)	b (p-value)	c (p-value)
PANSS_Total	$81.65(0)$	$-1.17 \times 10^{-1}(0)$	$-2.13 \times 10^{-4}(0.892)$
P_total	$19.37(0)$	$-3.70 \times 10^{-2}(0)$	$-3 \times 10^{-4}(0.592)$
N_total	$22.01(0)$	$-2.39 \times 10^{-2}(0)$	$-2.24 \times 10^{-4}(0.663)$
G_total	$39.90(0)$	$-5.62 \times 10^{-2}(0)$	$3.12 \times 10^{-4}(0.705)$

In this case, null hypothesis is "value of parameter being 0". From the above table, we can clearly see that for PANSS_score there is chance of 89.2% that c=0 and similarly for P_total, N_total, G_total 59.2%, 66.3%, 70.5%. So we say that, PANSS score is almost independent on treatment type. But this doesn't give emphasis as the above methods.

1.5 Conclusion

From the above methods we conclude that, Anonymized treatment is just working as good as standard medication and there is no improvement in overall sense. But when we consider negative symptoms only then newly proposed works well and for general symptoms standard medication worked well according to LOWESS plots. Even our Curve fitting method confirms this points with drop in probability as we have almost near 40%-50% chance for dependence of treatment on TxGroup is non-zero. We cannot comment strongly on individual improvement on each class of symptoms as we have almost similar results with minor deflections but one thing is sure that newly proposed treatment compared to existing has no effect in overall symptoms improvement.

2 Patient Segmentation

Question 2

Segment the schizophrenia patients into these k groups based on baseline measurements (where you decide on the value of k) and to describe each of the groups.

2.1 Pre-Analysis

Given dataset has around 39 features along with three additional features that we introuduced, making it total to 42. This is high dimensional dataset. But we can view atmost 3dimensions. So we have to abstract out all features in maximum of 3 dimensions using **dimensionality reduction**. As we are looking at baseline measurement we ignore visitDay field and as said earlier we will approximate individual scores with P_total, N_total, G_total (some sort of dimensionality reduction as we reduce from 30 dimensions to 3). Regarding considering fields like country, study etc will depend on the groups that we want to segregate data. I want to segregate data into groups to **identify the severity of initial symptoms**. So, I will ignore the other fields like Study (I am doing for all studies combined), ID's like raterId etc and, country, TxGroup due to less correlation with the measures. Now we are left with PANSS_score, P_total, N_total, G_total where we know that PANSS_score is sum of the other three and thus we ended up using P_total, N_total, G_total as our data to consider for clustering.

2.2 Segmentation

Now we figured out that we are using P_total, N_total, G_total features and classifying the initial groups of severity of Schizophrenia. Now we have to determine the value of number of clusters along with the method used for clustering. According to lecture-8 (in course), we have below approaches for clustering:

- Prototype methods
  - K-means
  - K-centers
  - D2-clustering
- Statistical modeling
  - Mixture modeling by EM algorithm
  - Model Clustering

All the plots along with code for this problem can be found here.

2.3 Statistical modeling Method - Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) are a probabilistic model that assumes a dataset is generated from a mixture of several Gaussian distributions. They can be fit using the expectation-maximization (EM) algorithm.

2.3.1 Deciding on value of K

As we are partitioning into K clusters we have to decide upon the optimal value of K needed. So we can use Akaike Information Criterion(AIC) and Bayesian Information Criterion(BIC) criterion as metrics for GMM.

Note Inertia or silhouette scores aren't reliable when the clusters aren't spherical or have different sizes. So we use a theoretical information criterion, such as BIC and AIC penalize models that have more parameters to learn (e.g., more clusters) and reward models that fit the data well.

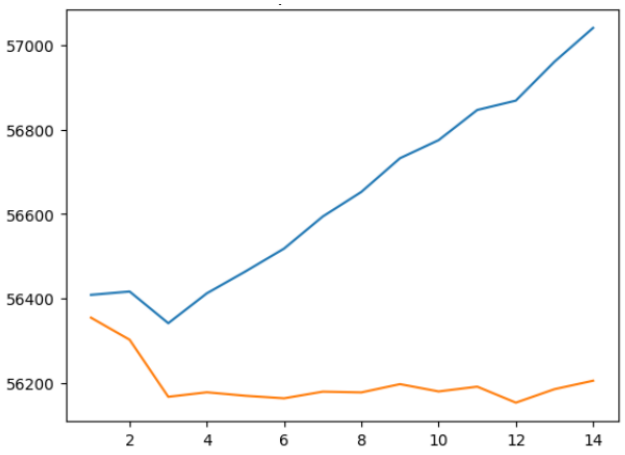


Fig - 5: AIC and BIC vs number of components (k) for GMM model

From the plots, we can clearly see that for minimum BIC we have number of components as 3 and for minimum AIC is 12 (difference is AIC of k=12 and k=3 are almost same too). Now we have conflicting between AIC and BIC criteria.

- Generally both AIC and BIC criteria give same value of k but if they contradict then we consider result with least BIC. (Reference - Jake Vanderplas Blog on GMM).
- The above is due to the fact that GMM does random initialization of inital parameters and AIC is more sensitive to initialization than BIC, so we have to opt for BIC criteria than AIC. (in case of conflicting values of k).

So we have the value of K_{opt} for GMM as 3 (using BIC). Now after plotting and labeling data using those three clusters we got the below 3D plot (3D because we chose 3 features).

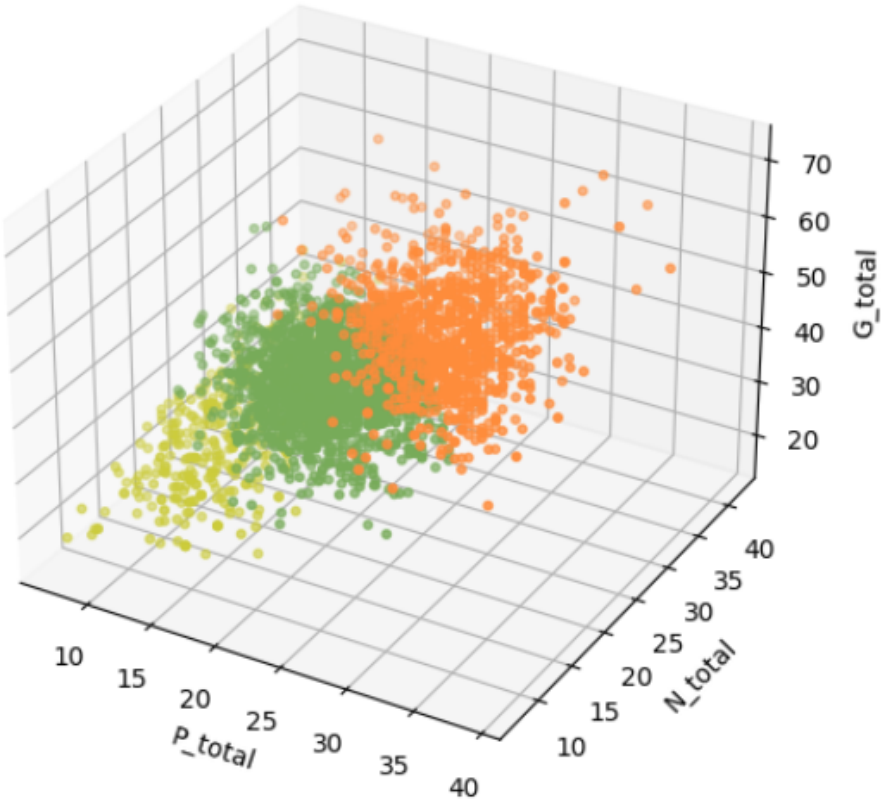


Fig -5: Figure showing 3 clusters of given data detected by GMM Model

2.4 Prototype Method - KMeans

KMeans algorithm aims to partition a set of observations into k clusters, where each observation belongs to the cluster with the nearest mean. The algorithm iteratively adjusts the cluster means until convergence.

2.4.1 Pre-processing

KMeans algorithm is sensitive to mean and variance so we have to normalize data before applying this algorithm. So we use **StandardScaler()** from scikit-learn to normalize all features of data and pass it further.

2.4.2 Deciding on value of K

As we are partitioning into K clusters we have to decide upon the optimal value of K needed. (Reference from 9th chapter of book "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow 2nd Edition").

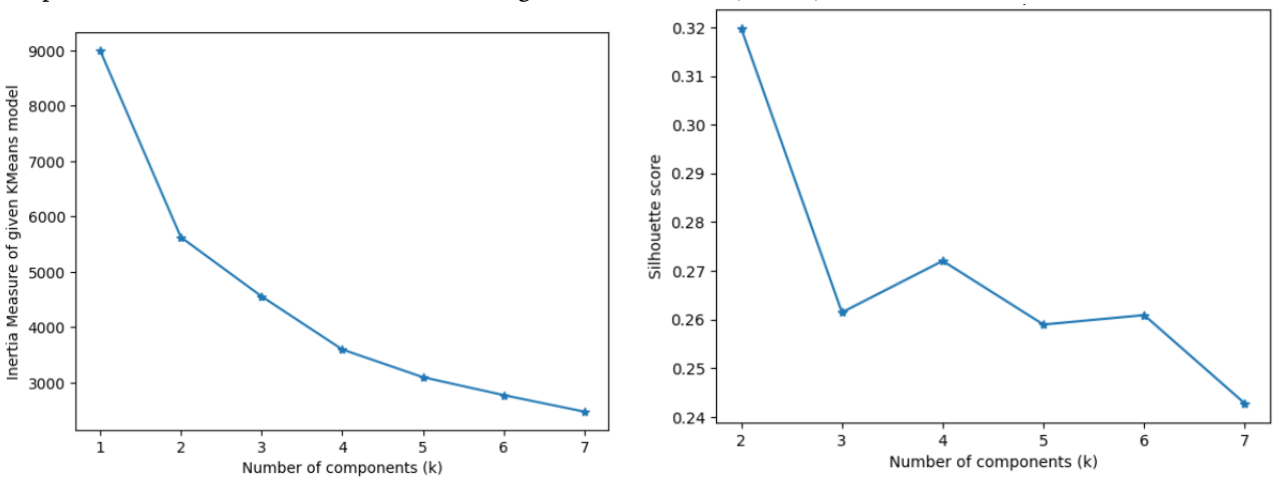


Fig - 6: (a) Inertia vs k (Elbow Method), (b) Silhouette scores vs k where k is number of components

- **Inertia plot** : For optimal value of K, we can consider metric called **inertia** which is the mean squared distance between each instance and its closest centroid. The one with lower inertia is optimal value of k, but from the above we have inertia decreasing down with value of k and thus we have to find an inflexion point called the **elbow**. Elbow is the value of lowest possible value of k where on increase in k wouldn't decrease inertia considerably (saturation phase). From the above curve using elbow method, we conclude that K_{opt} **= 4**.

- **Silhouette plot**: An instance's silhouette coefficient is equal to $\frac{(b-a)}{max(a,b)}$, where a is mean distance to other instances in same cluster (i.e., mean intra-cluster distance) and b is the mean nearest-cluster distance (i.e., the mean distance to the instances of the next closest cluster, defined as the one that minimizes b, excluding the instance's own cluster). When we take mean silhouette coefficient over all the instances we get **Silhouette score**. In general, one with higher silhouette score is better. From Fig-6 (b), we conclude that K_{opt} **= 2** for this method.

As silhouette is better than elbow method (reference), considering K_{opt} **= 2** and plotting clusters, gives below,

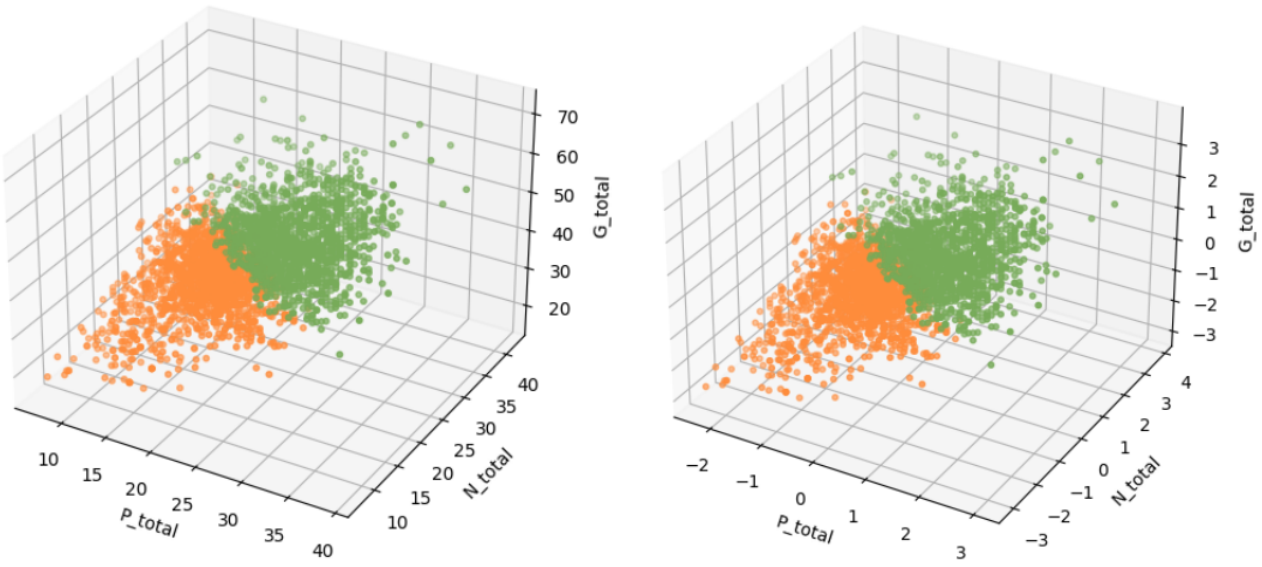


Fig -7: Clusters of data detected by KMeans (a) after feature scaling(initial values), (b) before scaling

2.5 Conclusion

For given data, GMM classified patients based on P_total, N_total, G_total into three categories as we consider BIC over AIC whereas KMeans classified patients into 2 categories only where we took Silhouette over Elbow method. Now the main question is, **what do clusters represent?**. As we have clustered using G_total, P_total, N_total our outcome would be saying about severity of Schizophrenia on patients on 0^{th} day of their visits.

- Gaussian Mixture Model has successfully classified the patients into three categories : Orange cluster representing highly effected, green being moderately effected and yellowish-green being least effected.

For GMM	Orange cluster	Yellowish-green cluster	Green cluster
Mean of each component	$(25.573, 24.994, 48.318)$	$(13.089, 20.126, 28.613)$	$(19.446, 21.705, 39.791)$

From the above table, orange cluster is shifted towards higher measures and then green, yellowish-green clusters. This proves that our hypothesis that orange is representing higher severe patients then green(moderate effect) then yellowish-green clusters (least effect).

- Doing the same for KMeans where it predicted two classes which can hypothesized as patients suffering Schizoprenia or not (in statistical sense).

For KMeans	Orange cluster	Green cluster
Centre of cluster after feature scaling	$(17.084, 20.572, 35.788)$	$(24.943, 25.015, 47.599)$
Centre of cluster for normalized data	$(-0.623, -0.414, -0.661)$	$(0.702, 0.467, 0.745)$

From the above table, green cluster is shifted towards higher measures and then orange. This proves that our hypothesis that green is representing patients diagnosed as Schizophrenia and orange for patients possibly not having noticeable effect of Schizophrenia.

3 Forecasting

Question 3

Predict the total PANSS score for the 18th- week assessment. Create a csv file that contains the PatientID and the predicted 18th-week PANSS score.

3.1 Approach

In this question we have to predict the 18th week score of the patients in Study-E using the data from Studies A to D. So based on the trends observed in patients in the previous studies we have to predict test subject's PANSS score.

- Just by looking at the problem one common solution could be Linear Regression. Let's see how did I modify it to perform better.
- As one can see there is huge data and features accross all the studies so, directly applying regression is meaningless. Considering all the data and features a faster method would be double exponential smoothing algorithm as we can see hints of time series application (score depends on days).
- Using that I trained the algo for 18th week prediction using studies A to D.
- There are other options too like Random Forest Reggressor or Support Vector Regressor but let's see how this simple algo works here.

3.2 Data Processing and Implementation

- I used the pandas library to read each file (A to D) as a data frame and concatenate them into one big data frame called train. It also reads another file as a data frame called test (E). Finally, I converted both data frames into numpy arrays, which are efficient for numerical computations.
- The featureExtractor function takes an input tensor (a multidimensional array) and a boolean flag called testSet. The function returns a numpy array that contains a subset of the input tensor. If testSet is False, it returns all the elements except the first seven and the last one. If testSet is True, it returns all the elements **except** the first seven.
- After using few other manipulations (which can be seen in code) I condensed all the data to two lists called X and Y.
- Then I used doubleExponentialMean that takes an input list of arrays and two parameters: a and gamma. The function implements a double exponential smoothing algorithm to make predictions based on the input data. The function loops over each array in the input list and initializes two variables which represent the predicted value and the gradient (or trend) respectively.
- It then updates these variables using a weighted average of the actual value and the previous prediction and gradient, where a and gamma are the smoothing factors. The function returns a list of predictions for each array in the input list.

$$\begin{aligned} l_t &= a * y_t + (1-a)(l_{t-1} + b_{t-1}) \\ b_t &= \gamma(l_t - l_{t-1}) + (1-\gamma)b_{t-1} \\ s_t &= l_t + b_t \end{aligned} \tag{2}$$

- The above equations are the basis for the double exponential a and gamma are the smoothing parameters for the level and trend components and t is the time index. The term "double exponential smoothing" refers to the use of two smoothing parameters, one for the level component and one for the trend component of the algorithm.
- The name "exponential smoothing" comes from the fact that the weights used to calculate the smoothed values decrease exponentially as we move further back in time
- After the processing is done and we calculate the best parameters using three metrics MSE (Mean Squared Error), MAE (Mean Absolute Error) abd RMSE. Although three metrics are not needed all three are generally used using the sklearn.metrics.functions.
- Then using these values I calculated the values of a and gamma for which RMSE is the least.

- Then using Linear Regression along with these best parameters from sklearn.linearModel fits the values on X and y which were populated in previous step. Then using predict I predicted the values of total PANSS score.
- As I used double Exponential method on X and y first then using linear regression it is somewhat a compound effect and combining the prediction from linear model and double exponential algorithm based on labels. (Can be seen in code)

3.3 Results

- After putting together a CSV file containng the predicted PANSS score of Patients of Study-E (18th week) I submitted the file to Kaggle for evaluation.

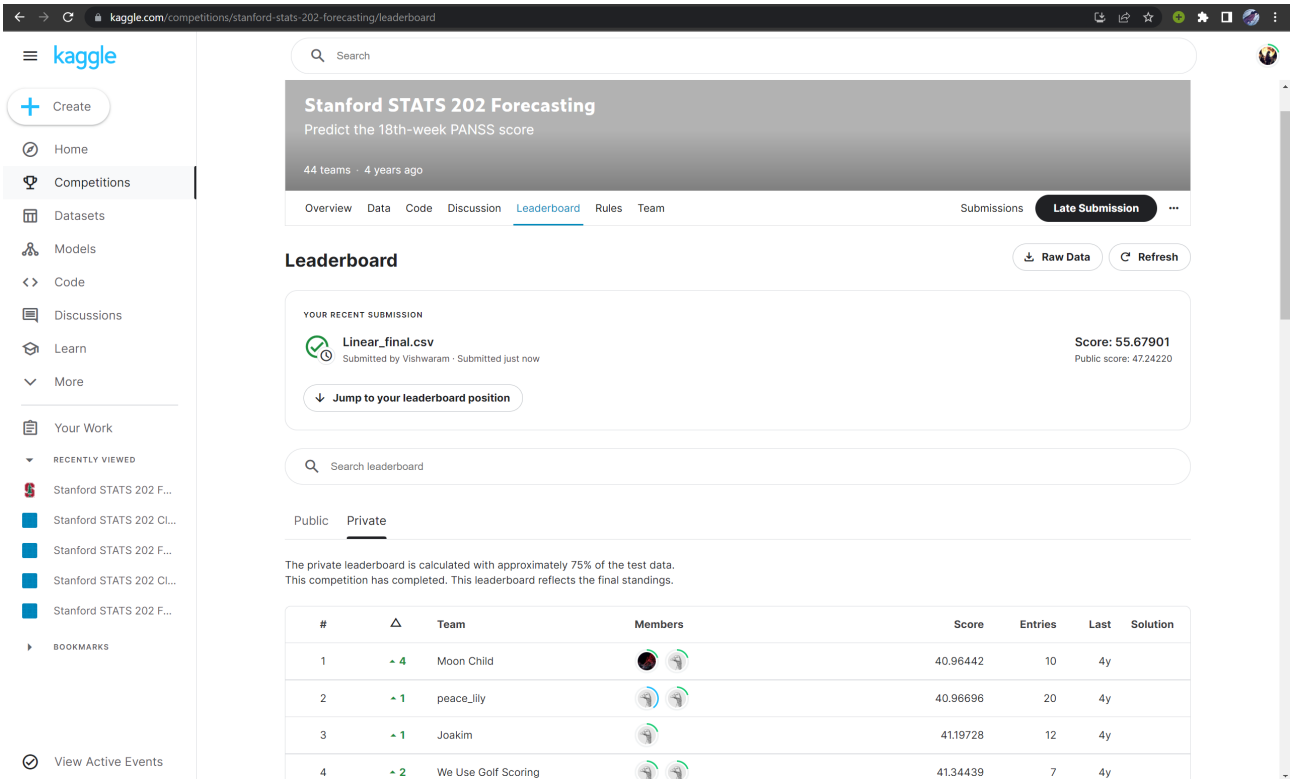


Fig -8: Kaggle submission results.

- For the predictions made using Linear Regression and Double Exponential Mean a score of 55.67901 was given by Kaggle obtaining 19th position out of 44 submissions.

3.4 Observations: Why Linear Regression works so well here?

In general Linear Regressor works well in the following conditions:-

- The relationship between the independent and dependent variables is linear.
- There is no high correlation among the independent variables.
- The observations are independent of each other.
- This goes without saying but if the relation between independent and dependent variables are linear.

The following observations can be made from Questions 1 and 2:-

-

Hence, Linear Regression works to a decent degree of accuracy in **this** case.

3.5 Conclusions and Improvement

In this problem as the data set followed a specific trend we were able to extract the use fullness of linear regression in combination with double exponential mean. But the general methods like Support Vector Regressor, Random Forest Regressor are always applicable which might be an improvement if we consider the optimal parameters.

4 Binary classification

Question 4

Predict which of the assessments in Study E will be either flagged for review or assigned to a CS. Create a csv file that contains the AssessmentID and the probability of the assessment being either flagged or assigned to a CS (Clinical Specialist).

4.1 Deep dive

In this question my objective is to find the probability that whether each patient in Study-E will get flagged (for detailed meaning of each term refer the Intro). For this I have considered Studies A to D as training data and Study E as test data. According to the original document this is the description of flagged/assign to CS :- "Examples include the patient assessment (as a whole) not making any sense, assessments that are inconsistent with previous ratings, and an outcome assessment trajectory that is infeasible. Consequently, clinical auditing firms are typically hired to validate the collected patient assessments. Assessments that are potentially erroneous are either flagged for review or assigned to a clinical specialist for follow up and confirmation."

4.2 Approach

Since any particular trends are not given to classify the patients to each group, I tried to study the data from A to D where the each patient was labelled and tried to predict the probability using a trained ML model. As it was hard to predict which columns are the exact features which determine the status of the patient I took all the columns from country to Lead status as all were affecting the final outcome. Then I used these columns to train my model for predicting the probability.

So, the basic version of this problem would be:- Given a patient which class does he/she belongs to? This version extends this and asks us to give us the probability that the person belong to that group.
Since its a classification problem the basic thought would be to use Logistic Regression, but owing to large amounts of data and a good number of features I think that wouldn't be a great model (We can use it but the data has to be heavily processed and I don't think the results would be that great..)

I proceeded to use decision tree based GradientBoostClassifier(GBC) for this task for reasons explained further.

4.3 Why GBC?

First let me give a brief intro to GradientBoostClassifier:- GBC is a ensemble learning method which is used to optimize both regression and classification problems. (We will be focusing on classification here)

- It combines several weaker models to create a strong predictive model.
- Here I'm using the implementation based on decision trees i.e, they are the weaker model.
- The algorithm builds an additive model in a forward stage-wise fashion and allows for the optimization of arbitrary differentiable loss functions. In each stage, regression trees are fit on the negative gradient of the loss function.
- I'm using sklearn's implementation and further information can be gathered from here

Now that we have some basic idea let's see it's advantages:

- Gradient boosting classifier can handle both numerical and categorical features and can perform implicit feature selection.
- Gradient boosting classifier is robust to overfitting and can achieve high accuracy with less data sensitivity
- Gradient boosting classifier can learn from the errors of previous weak learners and improve the predictions iteratively (well this applies to almost all ensemble learners but still I'm mentioning it here).
- Gradient boosting classifier can use different loss functions such as log loss to measure the performance of the mode which I used here for this task.

After all that I was convinced to use GBC on the data to which I fed all the columns after some processing discussed below.

4.4 Data Processing and Implementation

In this section I will explain what all data processing was done and how I used GBC to predict the probability. Please feel free to refer to the code attached for further understanding.

4.4.1 Data Preprocessing

- I collected the data from all studies that contain the scores and labels. I concatenated them into one array for training (Studies A to D) and another array for testing (Study E).
- I extracted the features and labels from the arrays and mapped the labels to numerical values using a dictionary. The dictionary labels "Passed" to 0, "Assign to CS" and "Flagged" to 1 respectively. This means that the classifier is performing a binary classification task, where 0 means the patient passed and 1 means the patient needs further attention.
- I used another function called Labels to create two lists: sequences and labels. The sequences list contains arrays of features extracted from the scores array for each patient. The labels list contains the numerical labels for each patient based on the dictionary. The function also assigns a label to each patient based on their last row in the scores array.
- I used another function called preprocess to pad each array in the sequences list with the edge values and then slide a window of size 2*windowsize + 1 over each array and flatten the window into a sequence. The function appends each sequence to an output list. The function does this to create more features for each patient based on their neighboring scores.

4.4.2 Model Training and Implementation

- I split the features and labels into training and testing sets with a 80-20 ratio (as this is the accepted ratio) and a fixed random state(for using 5 fold CV further) using the TrainTestSplit function from sklearn.
- Then I performed a grid search cross-validation to find the best hyperparameters for the gradient boosting classifier using a grid search cross-validation object. The object takes a classifier, a parameter grid, number of folds and a number of parallel jobs (I set this to max i.e -1 meaning it uses all available CPU threads) as inputs.
- The object performs an exhaustive search over all possible combinations of parameters in the grid and evaluates each combination using cross-validation on the training set. The object returns the best parameters and score based on accuracy.
- The ideal case should be using all the train data in the grid but as the data is huge (even after processing) along with the features, I ran the grid only on Study A and used those parameters on the rest of the studies to train the model. The parameters are as follows:- Best Hyperparameters: 'learningrate': 0.01, 'maxdepth': 4, 'nestimators':500 Note that these are only form study A 5 fold Cross Validation.
- I trained the gradient boosting classifier on the training set using the best parameters found by the grid search and evaluated and extended it for all the studies on both the training and testing sets using log loss as the metric. The classifier then makes predictions by taking a weighted vote of all the trees.

4.5 Results

Based on the method described above I was able to predict the probability using sklearn's predictProba and created the csv file for the evaluation and submitted on Kaggle Platform for verification/evaluation. With a Score: 0.66738 on test data from Kaggle my prediction stood in 30th position out of 44 with very near gap between each from 5th. The performance of the model could be improved but its time consuming and computationally intensive.

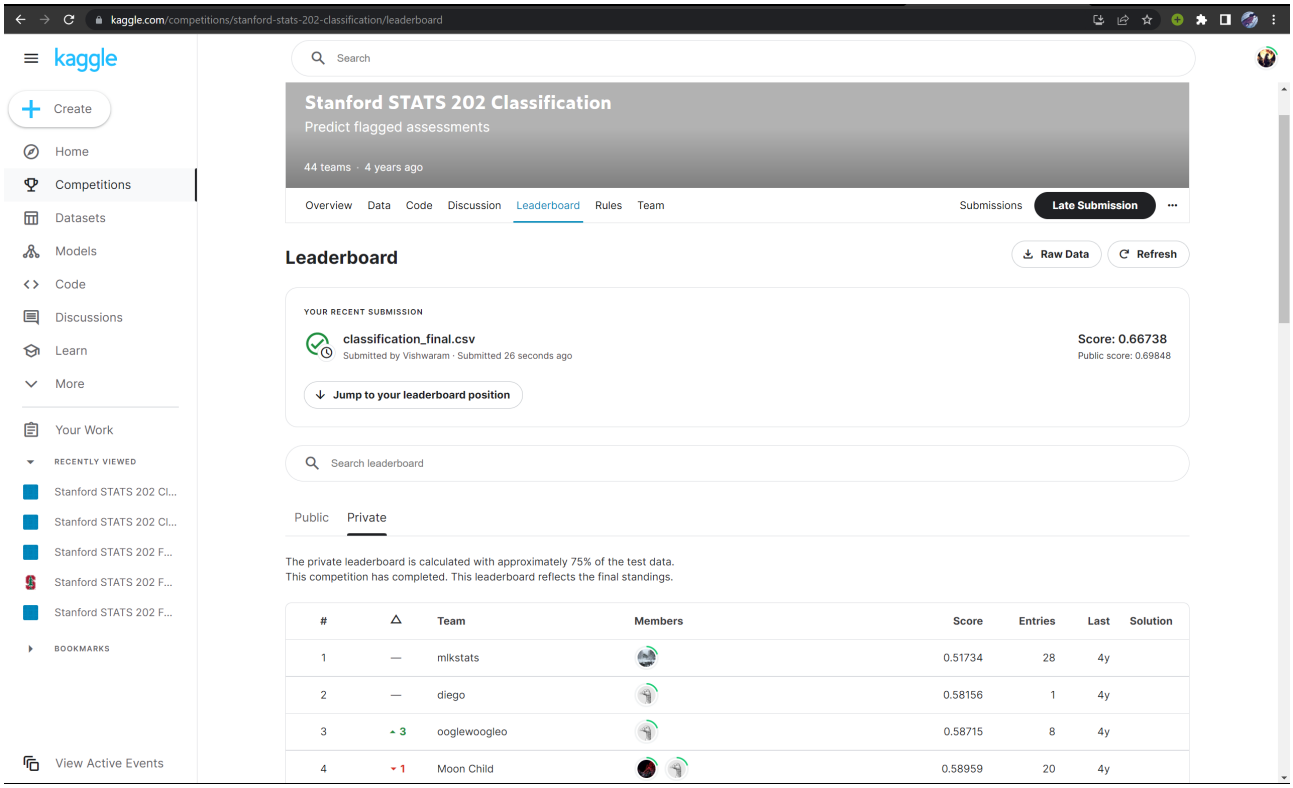


Fig -9: Kaggle submission results.

4.6 Conclusions and Improvement

So we can see that in the age of ML and Data Processing we use them for predicting the status without actually seeing the patient though it has some errors but we can get a rough idea on where and which patient we have to focus on. Though this is not the only method but one can always use different models and also can be estimated purely using statistics.