Project Report

1. Treatment Effect

1.1. Define Score

To evaluate treatment effect, a well-defined score is required to track the decrease of PANSS during a period of time. PANSS Decrease Percentage (PDP) is proposed in this project to track the total PANSS of a certain patient in a certain treatment period.

$$PDP = 1 - \frac{PANSS_{end\ of\ period}}{PANSS_{start\ of\ period}} \times 100\% \qquad (1)$$

where $PANSS_{start\ of\ period}$ refers to the PANSS of the patient's first visit (Day 0 in most cases), and $PANSS_{end\ of\ period}$ refers to the PANSS of the patient's visit on the date we are investigating (some day in 18th-week for example).

Note that PANSS are neither collected exactly every 7 days, nor always reliable for further use, so PANSS are sometimes missing for the day we are interested in. In this case, we take the PANSS of the visit closest to the day we are investigating.

1.2. Data processing

Data of all 5 studies are used as the dataset to evaluate treatment effect. The dataset is filtered as:

- Keep only assessments audited to be Passed.
- Keep only assessments of which the patient country is not ERROR.

1.2.1. Valid Patients Filtering

The first step is to find the valid patients and calculate their PDPs given a target day, the PANSS of which is about to be evaluated.

A valid patient is defined as:

- Number of passed assessments is large enough (>= `nvisits`);
- End day of all passed assessment is no early than the target day (>= `target_visit_day_threshold`);
- The start day of all passed assessment is not too far from Day 0 (<= `start_visit_day_threshold`).

Where `nvisits` and `target_visit_day_threshold` are by default 1 and 10, while `start_visit_day_threshold` will be assigned based on the target day we are to investigate.

By grouping the whole dataset by Patient ID and filtering the patients according to the rules above, we now have IDs of valid patients.

1.2.2. Calculate PDP

PDPs for all valid patients are calculated as below.

Filter the original dataset and keep only assessment records of valid patients, group the dataset by patient ID, retrieve the PANSS's of start day and target day for each patient, and consequently calculate the PDP as described in Equation (1).

Considering there are some noise in the time-dependent PANSS for each patient, we can smooth the curve using exponentially weighted moving average (EWMA) to get a more stable PDP score. However, I found there is not significant difference whether a smoothing is applied or not, so the PANSS of start and target day are simply selected from the original PANSS curve in this project.

1.3. Hypothesis Tests

The PDP data is now available when given a specific target evaluation day.

1.3.1. Evaluation of Treatment Effect

Take Day 120 as an example for treatment effect evaluation. The data frame of Day 120 is shown in Figure 1. The mean, standard deviation and variance are summarized in Table 1. It is very likely that the distribution of two groups are identical. Before moving on to a hypothesis test, it is necessary to check if PDP distribution meets the assumptions of hypothesis test. Additionally, as shown in Figure 2, both treatment and control group of Day 120 can be considered as normally distributed. Therefore, a Student's T-test is applied to see if control and treatment distributions are identical, implemented in python package `scipy.stats.ttest_ind`. The t-statistic and p-value is -0.017 and 0.987, respectively. Given a significant level of 0.05, the two distributions can be considered identical, indicating that treatment group shows no significant difference compared to control group.

PatientID	TxGroup	PANSS_decrease_pct
20009	Control	0.493
20010	Control	0.196
20013	Control	0.259
20016	Control	0.158
20018	Control	0.302
...	...	...
40038	Treatment	0.484
40048	Treatment	0.372
40057	Treatment	0.457
40120	Treatment	0.329
40158	Treatment	0.247

903 rows × 2 columns

Figure 1. A glance on data frame (in python) of PDP by Patient ID on Day 120.

Group	mean	std	var
Control	0.307	0.142	0.020
Treatment	0.307	0.136	0.018

Table 1. Statistics of PANSS for Control and Treatment Group on Day 120.

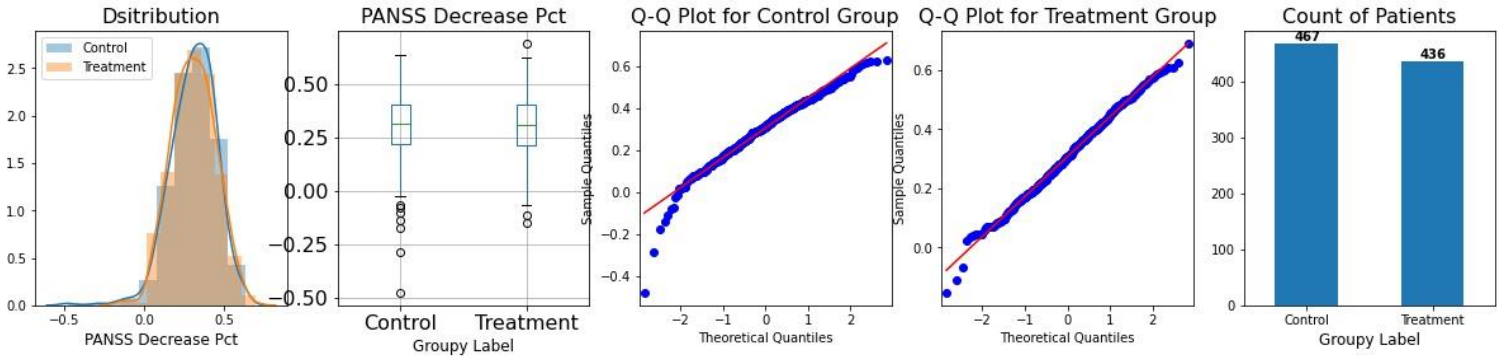


Figure 2. PDP Distribution Check for Treatment and Control Group on Day 120.

1.3.2. Time-dependent Treatment Effect

The treatment effect of medicals may vary over time, so it is essential to evaluate treatment effect on multiple days. Since there is no explicit explanation on treatment cycle, I take 120 days as a cycle (about 18 weeks) and evaluate on target days including 60, 120, 180, 240, 300, and 360.

	Control			Treatment			Hypothesis Test	
Day	mean	std	frequency	mean	std	frequency	t	p-value
60	0.201	0.155	643	0.204	0.150	598	0.357	0.721
120	0.307	0.142	467	0.307	0.136	436	-0.017	0.987
180	0.336	0.149	340	0.337	0.138	312	0.079	0.937
240	0.347	0.147	175	0.366	0.138	189	1.246	0.214
300	0.381	0.142	119	0.381	0.139	129	0.777	0.438
360	0.371	0.145	72	0.400	0.132	66	1.193	0.235

Table 2. Treatment effect over time.

It can be observed in Table 2 that, despite none of them show significant difference between two groups, p-value shows a decreasing trend over time. In other words, the long-term treatment effect of new medication may be better than that of standard medication. It is a research which is worth to conduct.

2. Patient Segmentation

For patient segmentation, the assessment scores on day 0 of each patient will be used as training dataset. Feature selection and Principle Components Analysis (PCA) will be used to reduce the dimension of feature space. Consequently, unsupervised clustering algorithms (K Means and Agglomerative Clustering) will be applied to conduct the segmentation. The result of clustering will be interpreted by extracting the characteristics of each patient cluster.

2.1. Data Preparation and Exploration

For patient segmentation, the dataset is filtered as below:

- Audit status is Passed.
- Patient country is not ERROR.
- Visit day is 0.

The filtered data is consequently grouped by Patient ID. Drop all columns irrelevant to patients (i.e. Rater ID, Site ID). We now have all the patients for segmentation.

There are 31 numerical columns in the dataset, 30 of which are PANSS ratings and one of which is the sum of 30 ratings. Out of 30 ratings of PANSS, 7 constitute a Positive Scale, 7 a Negative Scale, and the remaining 16 a General Psychopathology Scale. [1] It is obviously reasonable to include the sum of each category as a new column to the dataset. Besides, Composite score is also an essential part of PANSS system, which is a dipolar value defined by subtracting Negative from Positive. Composite score is not included in the dataset, so it is added as a new feature in this section. Now we have 35 numeric columns, 30 of which are PANSS ratings, 5 of are scores calculated from the 30 ratings.

Normality test and outlier detection can be conducted by plotting the distribution of all numeric features. Figure 3 shows the distribution of 5 calculated scores as the example of all 35 features. Neither extreme skewness nor outliers are observed.

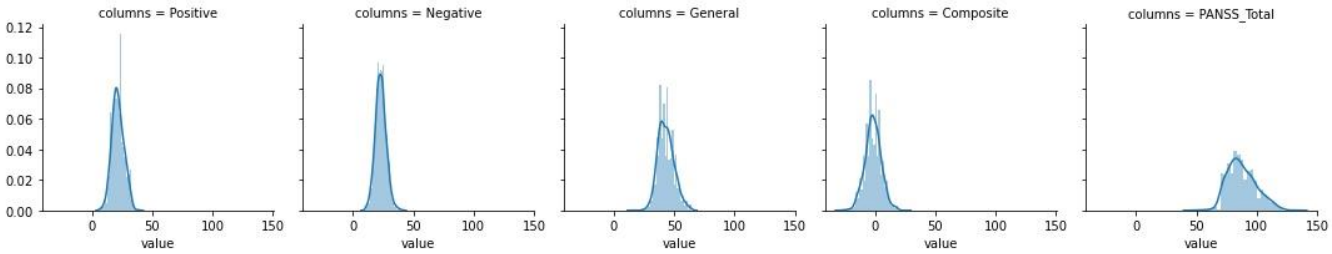


Figure 3. Distribution of PANSS scores.

2.2. Feature Selection and Dimension Reduction

2.2.1. Categorical Features

Dropping the irrelevant categorical features, there is only one left – patient country. Patient country seems a good feature for clustering, but it is extremely harmful to include it as a feature due to its high cardinality which leads to high dimensional sparse matrix generated by one-hot encoding. The high dimensional binary features will lead to the suppression of normalized numerical features and consequently misleads the model training.

As a result, only 35 numeric features are kept for clustering.

2.2.2. Numerical Features

In this step we need to decide which of the numeric features would be selected for clustering. It is practically a good choice to visualize the high-dimensional patient matrix to get an intuitive understanding on possible clusters. Scatter plots of patient scores are summarized in Figure 4. Figure 4(a) uses 3 features: Positive, Negative and General, while Figure 4(b) uses Composite and Total PANSS. Figure 4(c) uses the first two principle components generated by applying PCA on dataset, while Figure 4(d) shows the 2-D scatter plot generated by applying a dimension reduction algorithms named t-SNE, which is able to keep micro-structures during dimension reduction. [2] There is no recognizable clusters in Figure 4 (a)-(c), but in Figure 4(d), we can roughly see 3 clusters, lower left, upper center and lower right, surrounded by noise points. Note that t-SNE sometime leads to fake patterns, so hyperparameters perplexity and learning rate have been tuned to get the stable pattern.

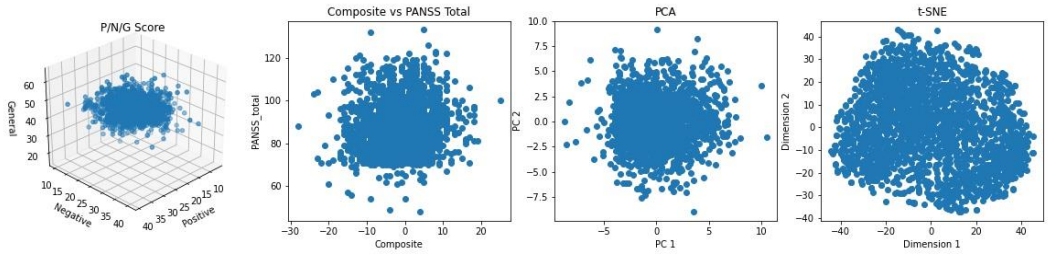


Figure 4. Scatter plots of patients based on feature combinations: (a) Positive, Negative and General; (b) Total PANSS and Composite; (c) first two principle components generated by PCA. (d) Scatter plot of 2-dimensional feature space generated by t-SNE.

Figure 4 indicates that it may not be a good idea to use all numerical features in clustering. Another option is to include only scores calculated from 30 ratings, namely Positive, Negative, General, Composite and Total PANSS. Despite information are lost when removing 30 ratings, we focused on the core information we need in evaluating a patient, which will lead to a much more efficient clustering as well as easy and clear interpretation.

Now we will compare the two option discussed above:

1. Include all numerical features (35 columns).
2. Include only calculated PANSS scores (5 columns).

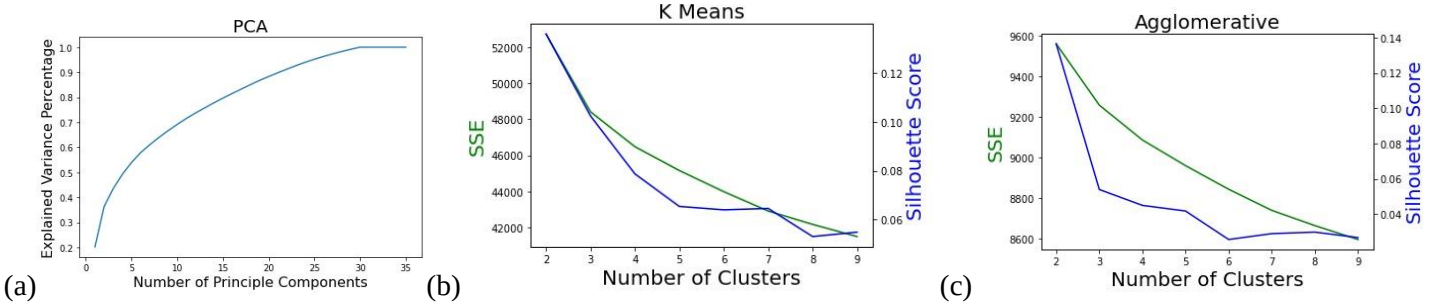
For each feature combination, the performance will be evaluate by the following rules:

- Number of clusters kept in PCA.
- Sum of squared distance (SSE) of K Means clustering result.
- Silhouette score of K Means clustering result.

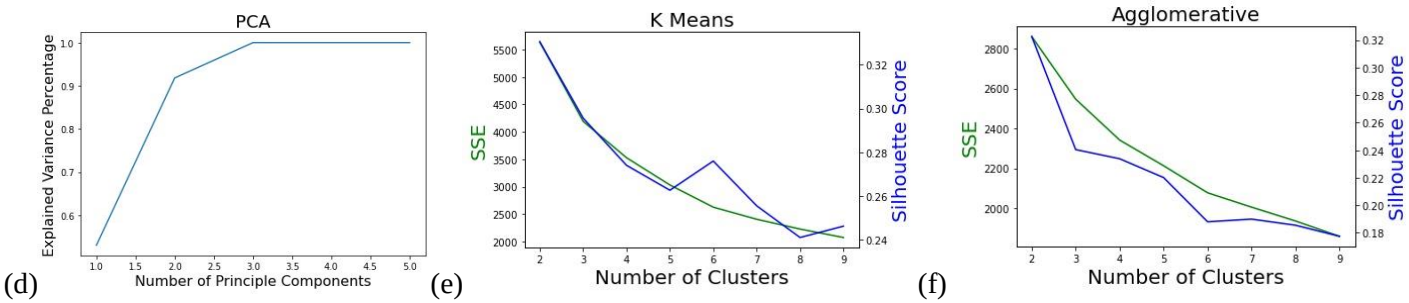
Therefore, it is better to select features together with clustering algorithms.

2.3. Model Selection and Performance

Two candidate algorithms are tested for clustering: K Means and Agglomerative Clustering. The metric used to measure is SSE and silhouette score.



(a) (b) (c)



(d) (e) (f)

Figure 5. PCA and clustering results of 35 numerical features (a-c) and 5 calculated PANSS scores (e-f).

The results of each combination of features and algorithms are shown in Figure 5, (a-c) for 35 features while (e-f) for 5 scores. When selecting 35 numerical features, no elbow point is observed in PCA, so 30 principle components are kept for 100% explained variance. In the consequent clustering, both K Means and Agglomerative Clustering indicates that the optimal number of clusters is 3.

When selecting 5 calculated PANSS scores as features, the performance gets better. Only 3 components are kept in PCA, keep 100% variance. And the consequent clustering show better silhouette scores around 0.30, compared to that around 0.10 using 35 features. Note that K Means suggests that the optimal number of clusters should be 6 rather than 3 due to a much lower SSE.

Based on the comparison shown in Figure 5, an optimal model for patient segmentation would be using 5 PANSS scores as the features and K Means (K=6) as the clustering model.

2.4. Segmentation Interpretation

To extract the characteristics of clusters, the dataset with 5 PANSS scores are grouped by cluster labels. The average of each feature are summarized in Table 3. To visualize the cluster characteristics, relative value plot and relative importance heat map are shown in Figure 6. Figure 6(a) shows the relative value of each cluster given a specific feature. The relative value is calculated by normalizing the dataset of patients, followed by grouping by cluster label and getting the average. It can be observed, for instance, Cluster # 0 shows the largest Composite while Cluster # 2 and # 3 show the lowest Composite. Similarly, relative importance is to measure how far an average of a feature is from that of the population, which is calculated by subtracting the average of population from that of the feature and minus 1.

Table 3. Summary of feature means for each cluster and the population before clustering.

Cluster #	***Positive (average)***	***Negative (average)***	***General (average)***	***Composite (average)***	***Total PANSS (average)***
0	26.77	17.95	42.75	8.82	87.47
1	21.87	23.35	44.99	-1.48	90.22
2	22.46	30.62	49.84	-8.17	102.92
3	15.66	25.31	39.02	-9.66	80.00

4	29.02	25.51	52.52	3.51	107.05
5	19.30	19.37	37.20	-0.07	75.86
population	21.61	23.10	43.17	-1.49	87.88

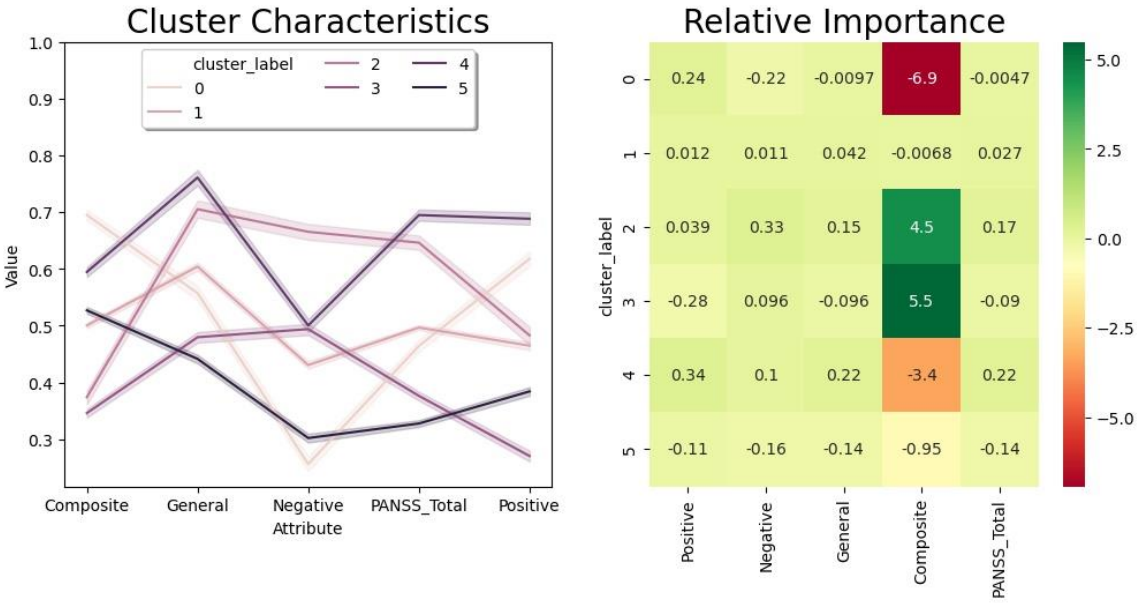


Figure 6. Cluster Characteristics shows the relative value of each PANSS score. Relative importance shows how far the average of each feature varies from that of the population.

Based on the results shown in Figure 6, the characteristics of each cluster can be easily extracted and summarized as shown in Table 4. Cluster #1 shows no significant difference compared to population before clustering, while the others are marked high or low in at least one PANSS score field. The interpretation provides insights on patient groups before the treatment, which can be used as a new feature when conducting other analysis tasks.

To predict cluster label for new incoming patients, an easy and fast way is to quantize the labels Low and High into a specific interval using the normalized relative values shown in Figure 6(a), so that new incoming patients can be labeled according to their PANSS scores without doing the clustering again. Despite a trained K Means is able to predict labels for new patients, most clustering algorithms are not. Therefore, a more general way to assign clusters, and precise enough in most cases, is to use a threshold or interval extracted from cluster characteristics.

Table 4. Characteristics of Clusters.

Cluster #	***Positive***	**Negative**	**General**	**Composite**	**Total PANSS**
0	High	Low	-	High	-
1	-	-	-	-	-
2	-	High	High	Low	High
3	Low	-	-	Low	-
4	High	-	High	-	High

5	-	Low	Low	-	Low

3. Forecasting

3.1. Data Processing

3.1.1. Train Data

Train data consists of all data in Study Group A – D.

Step 1. Get valid patients.

Since we need patients whose assessment history is longer than 18 weeks, we define, similar to what is done in Section 1.2.1, valid patients as

- `start_visit_day_threshold = 6,`
- `target_visit_day_threshold = 18 * 7,`
- `nvisits = 3,`

to keep only patient whose last visit covers 18th week and first week records is not missing. `nvisits = 3` is assigned to enable the interpolation which will be detailed in Step 4.

Step 2. Convert `VisitDay` into `Week`.

For example, day between 0 ~ 6 is converted to week 1; day between 7 ~ 13 is converted to week 2.

Step 3. Get assessment history for each patient.

Use pivot table to get a table, whose index is patient ID, columns is week number and value is total PANSS. If multiple values are found for the same patient and week combination, take the average. Now we have the PANSS history of each valid patient of the first 18^{th} weeks, despite that many values in the data frame is missing due to the unevenly spaced time-series PANSS values.

Step 4. Interpolation.

To train a forecasting model, all PANSS values of first 18 weeks are required. The first 17 weeks are features, while the 18^{th}-week is the target, so interpolation is applied to impute missing values for the data frame generated in Step 3. Spline interpolation is a simple and effective method to convert unevenly spaced time series into evenly spaced. As shown in Figure 7, linear spline interpolation and quadratic spline interpolation are applied on PANSS time-based sequence of Patient 30140. Both interpolations seem good enough for imputation. In model selection section we will decide which one to use. Note that quadratic spline required at least 3 data points to start, which is why `nvisits` is assigned to 3. If higher order spline is applied, `nvisits` should be adjusted accordingly.

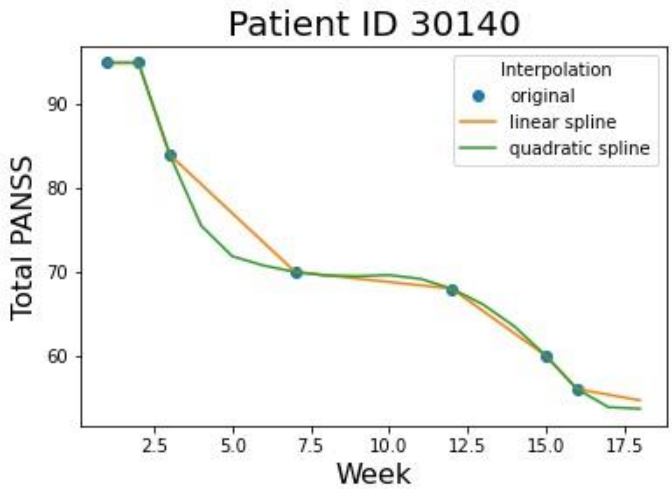


Figure 7. Interpolation for Patient # 30140. Blue dots are original data points, while colored lines are interpolation.

3.1.2. Test Data

Test data consists of all data in Study Group E.

The process is different from train data because we do not drop any observation is test data. A predicted values is required for all test data. It is not wise to drop the patients who visited less than 3 times. Therefore, the test data will be split into two groups based on how many time they visited in first 18 weeks and will be processed and modeled differently.

Step 1. Similar to what is done for train data. We convert VisitDay to Week and get assessment history using pivot table, without any identifying if the patient is valid.

Step 2. Interpolation method is different from training. First 17 weeks, instead of 18 weeks, is required. Since most assessment history of patients in test data does not covers the first 17 weeks, which means interpolation is not able to fill all missing values (tailing missing values may exist due to the last visit is ahead of 17^{th} week), linear-spline extrapolation is used to fill the tailing missing values behind the last visit. Given the interpolated PANSS time series, a linear regression is good enough to predict the 18^{th}-week PANSS, since I observed the PANSS history shows a linearly decreasing trend averagely, as shown in Figure 7.

Step 3. Note that quadratic spline requires at least 3 visits, so Step 2 is designed for patients who visited no less than 3 times. For patients who visited less than 3 times, their assessment history is hardly time series. Therefore, their 18^{th}-week prediction will be fitted by averaging the 18^{th}-week PANSS's of those whose first-visit PANSS is similar using K Nearest Neighbours (KNN) regression model. For example, Patient X has a first-visit PANSS of 80, and never visits after that. To predict his 18^{th}-week, we find the 3 patients who first-visit PANSS is 80, 81 and 82, whom we called his 3 nearest neighbours. Averaging the 18^{th}-week PANSS of the neighbours, we get the prediction for Patient X's 18^{th}-week PANSS.

3.2. Model Selection

3.2.1. Linear Regression

As discussed, interpolation methods will be selected between linear spline and quadratic spline, while linear regression model for patients who visited no less than 3 times will be selected from

linear regression with different regularization, namely linear regression, lasso regression and ridge regression. The results are summarized in table 5. Note that the regression performance is evaluated based on interpolated target, instead of ground truth, which is missing in most cases in our train set, the results of regression is actually measuring how good the model fits the interpolated values.

Table 5. Model selection. Evaluate each combination of interpolation and regression.

Model Selection		Train Metric		Test Metric	
Interpolation	Regression	MSE	R^2	MSE	R^2
Linear Spline	Linear	4.1598	0.9612	4.4594	0.9514
	Lasso	4.2082	0.9607	4.3601	0.9525
	Ridge	4.1598	0.9612	4.4573	0.9514
Quadratic Spline	Linear	1.1230	0.9902	1.2406	0.9875
	Lasso	1.4464	0.9873	1.5513	0.9841
	Ridge	1.1287	0.9901	1.2351	0.9876

It can be observed in Table 5 that ridge regression on quadratic-spline interpolated data shows the best performance: high R^2 and low MSE for both train and test set (indicating low bias), small gap between train and test performance (indicating low variance).

Hyperparameters are tuned to get an optimal result for regression models in Table 5. The penalty coefficient of Lasso and Ridge is assigned as 0.1 and 1.0, respectively. 5-fold cross validation is applied when evaluating each model.

Therefore, quadratic spline interpolation combined with ridge regression is the model to go.

3.2.2. KNN Regression

As a supplement, KNN regression are applied to make prediction for patients who visited less than 3 times in test set. For simplification K is assigned as 3, since what I want here is simply averaging 18^{th}-week PANSS's of the nearest neighbors. Euclidian distance is used in KNN, which is equivalent to Minkowski in our 1-dimension case. The average is calculated by the sum weighted by the inverse distance between an observation and its neighbor.

3.3. Interpretation

Since all 17 features are total PANSS's, recalling not required, the coefficients of ridge regression can be interpreted to be the importance of the feature. As shown in Figure 8, it can be observed that PANSS assessed in a week closer to 18^{th} week trend to have higher impact on prediction, which make great sense. In other words, 18^{th}-week PANSS depends mainly on 10th – 17^{th} week scores.

The prediction on Study E achieved a log loss of 8.56433 as shown on leaderboard of corresponding Kaggle competition.

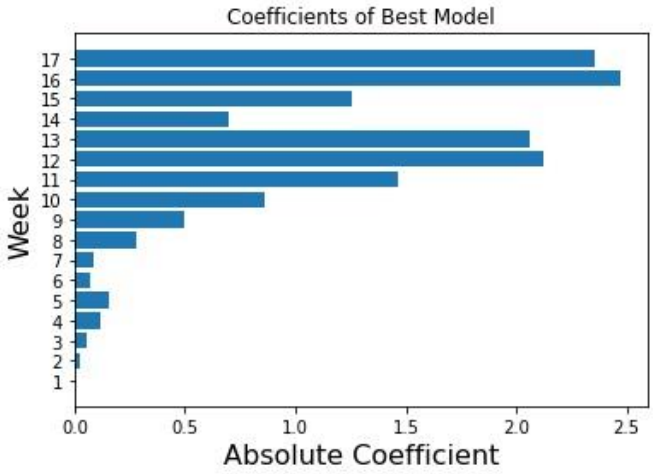


Figure 8. Absolute values of coefficients in optimal model of ridge regression.

4. Classification

Classify if an assessment record is reliable or not. Train set consists of all records of Study A – D, test set consists of all records of Study E.

4.1. Data Processing

4.1.1. Feature Selection

Drop all columns which are not relevant to an assessment result, such as Study, Country, Patient ID, Site ID, Rater ID, VisitDay. Assign Assessment ID as index of the data frame. As patients are randomly selected to treatment and control group, TxGroup is irrelevant when identifying if an assessment is valid.

As a result, 30 ratings and total PANSS are kept to make classification.

4.1.2. Feature Engineering

First of all, it is necessary to include Posotive, Negative, General and Composite as new features to get the outline of a assessment.

Secondly, it is necessary to consider the basic patterns of invalid assessments. As mentioned in the project instruction, there are some patterns which indicates an invalid assessment:

1. Patient assessment as a whole not making any sense (e.g. some scores indicating complete psychosis while others indicating complete normality),
2. Assessments that are inconsisent with previous ratings (e.g. scores are completely reversed),
3. An outcome assessment trajectory that is infeasible (e.g. scores indicating that a patient has complete psychosis, is subsequently normal, and has complete psychosis again).

All 3 patterns are essentially similar because they focus on the consistence of assessment score time series. For example, (1) a total PANSS, like 100, 80, 100, 80, 100, 80, make no sense because it goes up and down and no stable result can be observed; (2) a total PANSS series like 100, 110, 120, 130, 140, the scores of which keeps going up, showing completely reversed compared to the average treatment effect as we found when evaluating treatment effect. To detect such patterns as described, the time series of PANSS scores or ratings will be fitted by a simple

linear regression, which provides its metrics, like MSE, R^2 and slope, to detect the patterns mentioned above. Example (1) will be detected due to a high MSE and low R^2, while example (2) will be detected due to a positive slope.

Another feature can be useful is the previous value in time series. For example, a patient has a total PANSS of 80 in an assessment, but in the next assessment he/she got a total PANSS of 100. The second score may be labeled as invalid. To detect pattern like that, a new feature of previous value will be powerful if the pattern does exist. For the first score which does not have a previous neighbor, assign the values of itself.

To summary, the 3 metrics of linearity will be added as new features for the following scores: Positive, Negative, General, Composite and Total PANSS, which sums up to be 15 new features. Additionally, one new feature named previous neighbor score is also added.

4.2. Model Selection

Random Forest (RF) is an efficient model for binary classification. Compared to other classification models like support vector machine (SVM), the most important advantage is easy to interpret and embedded feature selection due to the feature importance calculated by decision tree model.

Hyperparameters are tuned as shown in Table 6. Note that the train set consists of 15841 valid assessment and 5104 invalid ones, indicating imbalanced classes. It is necessary to assign class weight so that the minority class will not suffer from low precision and recall. 3-fold cross validation is applied along with parameter tuning to evaluate the log loss of each candidate model.

Table 6. Hyperparameter tuning on Random Forest Classifier

Hyperparameter	**Candidate Values**	**Optimal Value**
Max Depth of a single tree	10, 12, 14	12
Max Feature used in a single tree	sqrt(p), p/2, where p is number of all features	sqrt(p)
Min sample to form a leaf	4, 8	8
Min samples to split a node	5, 10	10
Number of trees	100, 200	100
Class weight	balanced, balanced_subsample	balanced

4.3. Performance and Interpretation

Performance of the best model on validation set is summarized in Table 7, evaluated with precion-recall-f1 system. It can be observed that both majority and minority class has f1 score over 0.8, despite that minority class is a little lower than majority class.

Feature importance are calculated based on gini impurity of each tree in random forest. The top 20 out of 63 features are shown in Figure 9. It is obvious that 18 out of 20 of them are features designed in feature engineering, which strongly supports the pattern detection strategy I proposed above. The most important features are those that measure the consistence of assessment sequence. 5 scores, Positive, General and Composite, are proved important in classification, which makes great sense. Surprisingly, G12 stands out of all 30 ratings to be the 14^{th} important feature. G12 refers to lack of judgement and insight, which is defined as impaired awareness or

understanding of one's own psychiatric condition and life situation. [1] It indicates the auditors who audited the dataset may have tends to label based on exception of G12.

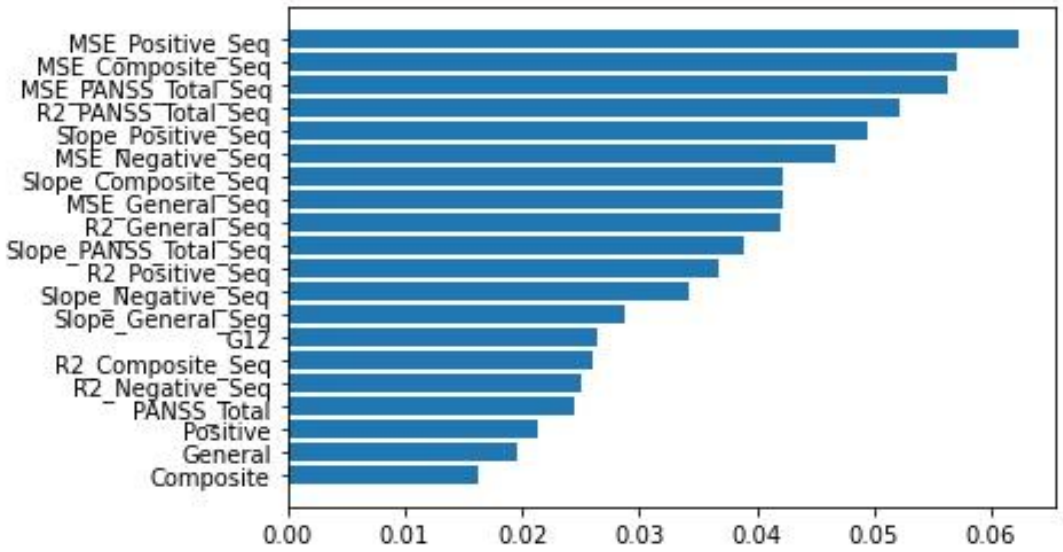


Figure 9. Feature Importance of Random Forest Classifier

Table 7. Precision, recall and f1 score

Class	Precision	Recall	F1 score	Number of Observations
Valid (0)	0.94	0.94	0.94	15841
Invalid (1)	0.82	0.82	0.82	5104

References

[1] Stanley R. Kay, et al, SCHIZOPHRENIA BULLETIN, Vol 13, No 2, 1987.

[2] Laurens van der Maaten, https://lvdmaaten.github.io/tsne/