國立臺灣大學
National Taiwan University

資料科學計算 Final Project

姓名：鄭書承

學號：R10246006

系級：應數所碩一

2022 年 1 月 3 日

資料科學計算 Final Project

鄭書承

1 Data Observation and Preprocessing

The training data set of GDSC contains 866 cell lines and 16190 features, and the test data set of patients, it contains 25 and 16190 features.

Due to the problem that the number of features is much lager than cell lines, we could not fit the model properly. Hence, we need to do the features selection to find out the real important features influencing whether the drug is sensitive or resistant.

Last but not least, we need to check whether the number of sensitive and resistant to the drug is balance or not. From the figure 1, we can see that the number of sensitive data is 680 and the number of resistant data is 186, which is imbalance, so we need to balance our data.

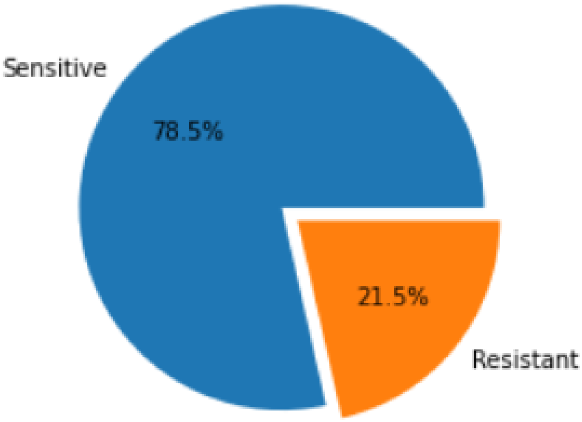


Figure 1: The section of each sensitive and resistant data

The imbalance would cause the model always guess the same answer like the situation of figure 2.

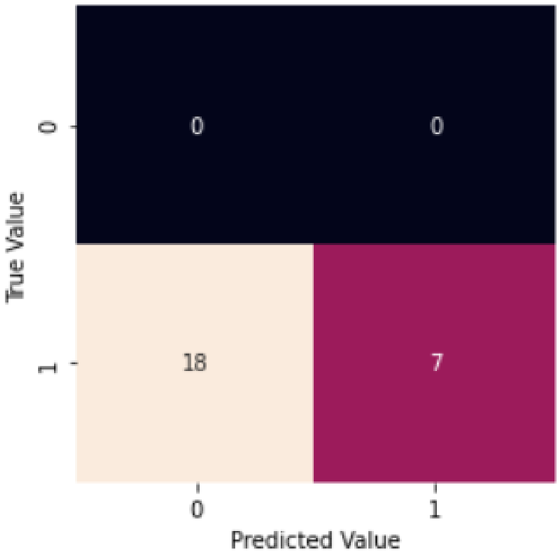


Figure 2: The confusion matrix of Lasso regression with the original data

There are two main idea to solve imbalance data which are over-sampling and under-sampling. Hence, I simply compare the result of this two idea by SMOTE and ClusterCentroids to construct new data. Moreover, compare them by predicting the test data after fitting the new data constructed in different methods with the model that selects the important features by Lasso regression. However, to compare the result, we need to define the score function. Due to the expensive cost of targeted therapy, I think the primary goal of the model is to find the patients who are truly sensitive to the drug and decrease the false positive rate. We also know that the definition of the f_β score function is

$$f_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}.$$

For satisfying the criterion above, I use the f_β score function with $\beta = 0$, which is exactly precision rate of the data to be my score function. From the figure 3 and figure 4 below, the f_β scores of the Lasso regression with the data constructed by SMOTE and ClusterCentroids are 0.25 and 0.5 respectively. The result may be caused by creating the wrong new data by SMOTE, so I decide to under sample the data by ClusterCentroids.

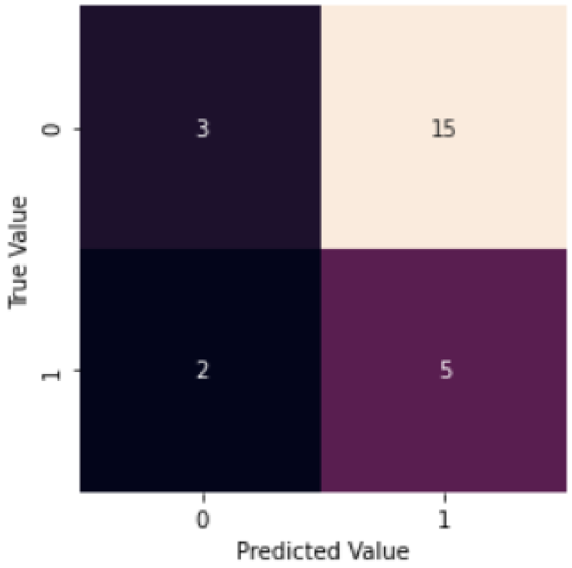


Figure 3: The confusion matrix of Lasso regression with the data constructed by SMOTE

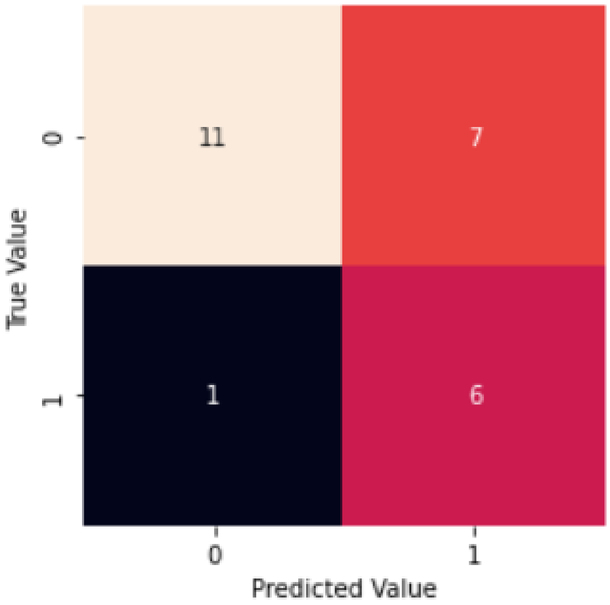


Figure 4: The confusion matrix of Lasso regression with the data constructed by ClusterCentroids

2 Features Selection

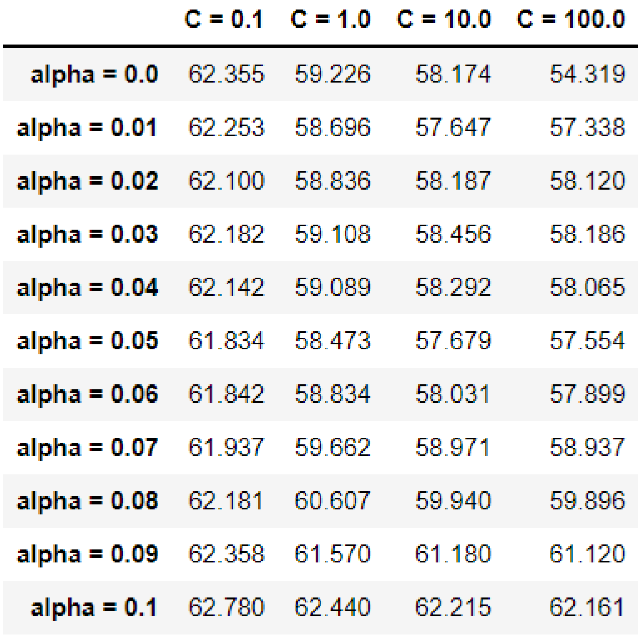
Lasso, Anova, PCA are famous way to do a feature selection, so I am interested in comparing these ways. Lasso is generated by linear regression with the penalty of variable numbers in L^1 such that it make some covariates converge to 0, which means they are not importatnt to the model. Moreover, AONVA is also generated by linear regression and it choose the covariates by doing t-test to check whether they are significant for model. The last method compared is PCA, which decide model by finding principal components to extending new axes with the largest variance.

2.1 Lasso

First, I select the 12 features by Lasso regression and make a logistic regression model fitted by those features. Therefore, the hyperparameters above are the wights of L^1 penalty α in Lasso regression and the weights of L^1penalty and make the hyperparameters space as the followings:

$$\begin{cases} \alpha = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 \\ C = 0.1, 1, 10, 100 \end{cases}$$

To find the best hyperparameters in the above space we assume, we need to do a repeated cross validation, which splits the training data in 5-folds and repeat them in 100 iterations to make sure the consequence is robust. From the sheet below, we can see that 62.780 is the highest average cross validation score in this 100 iterations, so we set our hyperparamters $\alpha = 0.1$, and $C = 0.1$.



	C = 0.1	C = 1.0	C = 10.0	C = 100.0
alpha = 0.0	62.355	59.226	58.174	54.319
alpha = 0.01	62.253	58.696	57.647	57.338
alpha = 0.02	62.100	58.836	58.187	58.120
alpha = 0.03	62.182	59.108	58.456	58.186
alpha = 0.04	62.142	59.089	58.292	58.065
alpha = 0.05	61.834	58.473	57.679	57.554
alpha = 0.06	61.842	58.834	58.031	57.899
alpha = 0.07	61.937	59.662	58.971	58.937
alpha = 0.08	62.181	60.607	59.940	59.896
alpha = 0.09	62.358	61.570	61.180	61.120
alpha = 0.1	62.780	62.440	62.215	62.161

Figure 5: The sheet of hyperparamters α and C

Next, we fit the model by whole training data and predict whether the patients in test data are sensitive or resistant to this drug. From the figure 7, result of ROC curve, we can see that the accuracy of the the model is around 0.770, and the f_β score is around 0.462.

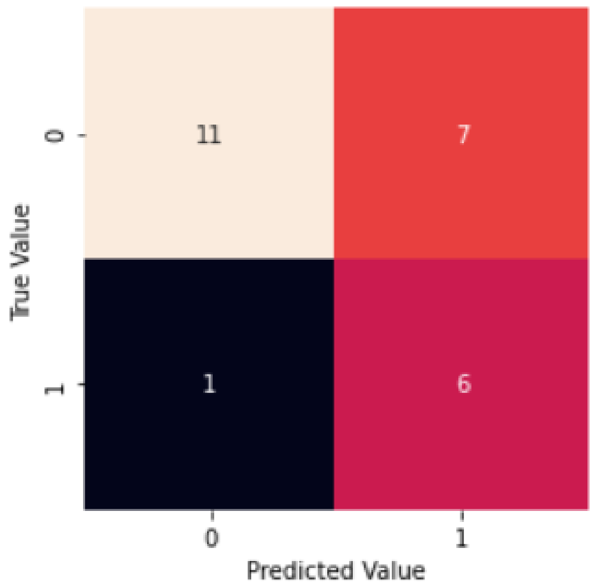


Figure 6: The confusion matrix of Lasso regression with the data constructed by ClusterCentroids

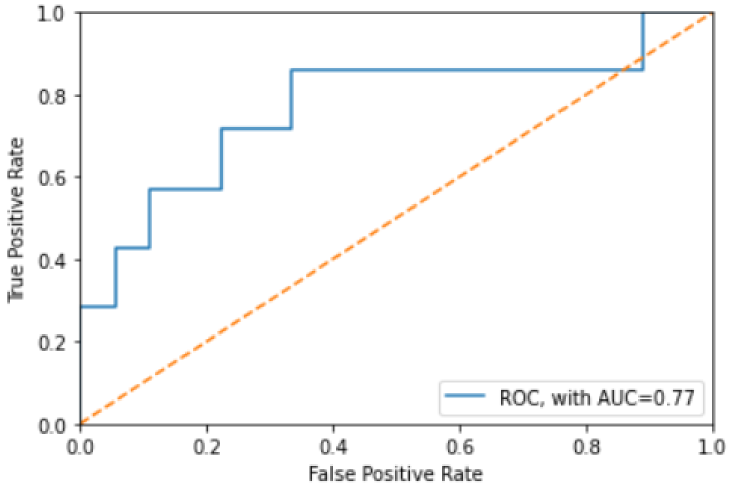


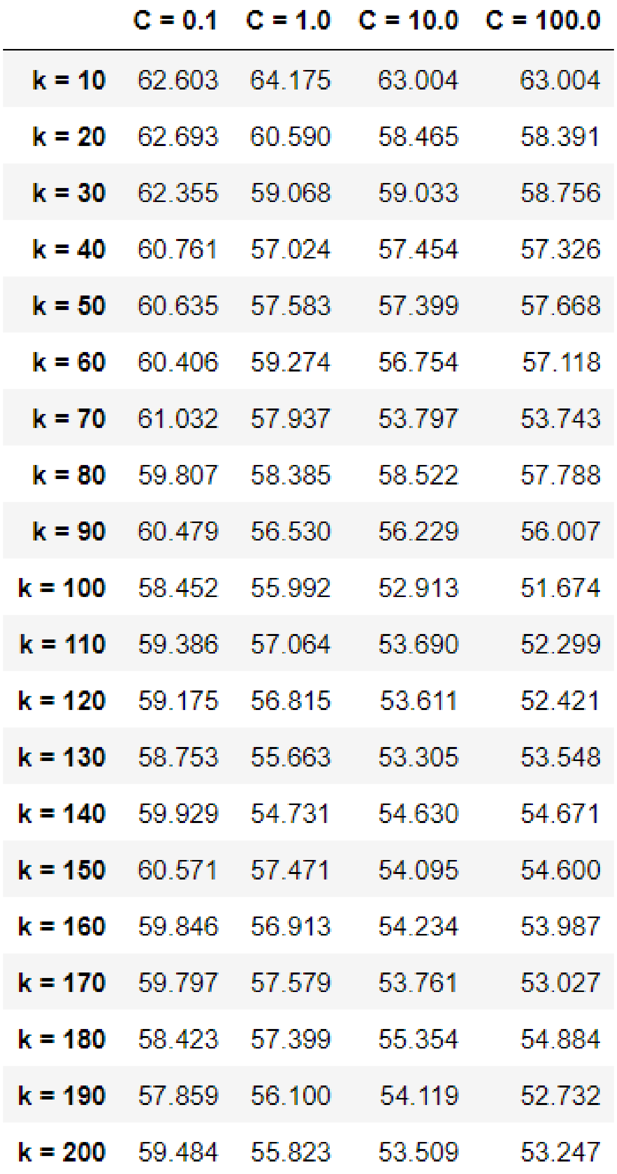
Figure 7: The ROC curve of Lasso regression with the data constructed by ClusterCentroids

2.2 Anova

Next, I select the 10 features by Lasso regression and make a logistic regression model fitted by those features. Therefore, the hyperparameters above are the the number k of features selected by Anova and the weights of L^1penalty and make the hyperparameters space as the followings:

$$\begin{cases} k = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200 \\ C = 0.1, 1, 10, 100 \end{cases}$$

Again, to find the best hyperparameters in the above space we assume, we need to do a repeated cross validation, which splits the training data in 5-folds and repeat them in 100 iterations. From the sheet below, we can see that 64.175 is the highest average cross validation score in this 100 iterations, so we set our hyperparamters $k = 10$, and $C = 1$.



	C = 0.1	C = 1.0	C = 10.0	C = 100.0
k = 10	62.603	64.175	63.004	63.004
k = 20	62.693	60.590	58.465	58.391
k = 30	62.355	59.068	59.033	58.756
k = 40	60.761	57.024	57.454	57.326
k = 50	60.635	57.583	57.399	57.668
k = 60	60.406	59.274	56.754	57.118
k = 70	61.032	57.937	53.797	53.743
k = 80	59.807	58.385	58.522	57.788
k = 90	60.479	56.530	56.229	56.007
k = 100	58.452	55.992	52.913	51.674
k = 110	59.386	57.064	53.690	52.299
k = 120	59.175	56.815	53.611	52.421
k = 130	58.753	55.663	53.305	53.548
k = 140	59.929	54.731	54.630	54.671
k = 150	60.571	57.471	54.095	54.600
k = 160	59.846	56.913	54.234	53.987
k = 170	59.797	57.579	53.761	53.027
k = 180	58.423	57.399	55.354	54.884
k = 190	57.859	56.100	54.119	52.732
k = 200	59.484	55.823	53.509	53.247

Figure 8: The sheet of hyperparamters k and C

Next, we fit the model by whole training data and predict whether the patients in test data are sensitive or resistant to this drug. From the figure 10, result of ROC curve, we can see that the accuracy of the the model is around 0.635, and the f_β score is around 0.308.

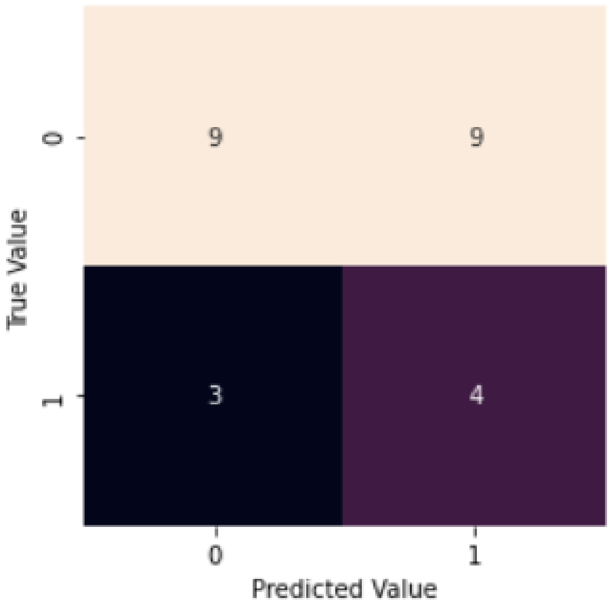


Figure 9: The confusion matrix of Anova with the data constructed by ClusterCentroids

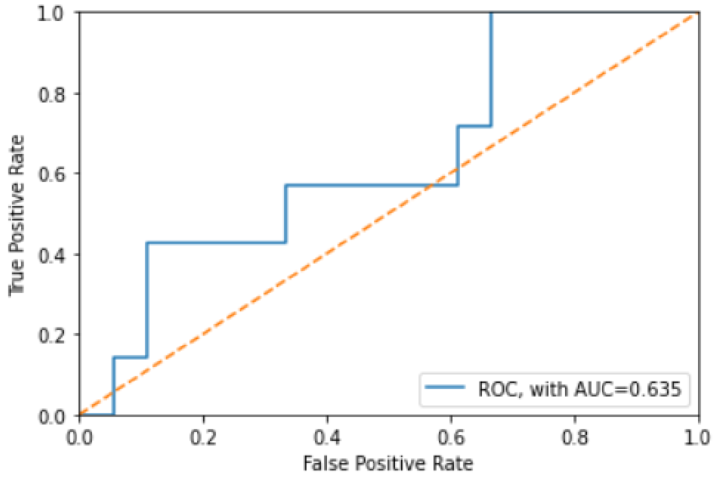


Figure 10: The ROC curve of Anova with the data constructed by ClusterCentroids

2.3 Stacking

I stack the two model above by logistic regression model like figure 11 to expect the logistic regression in level2 can distinguish which answer from the model in level1 is correct.

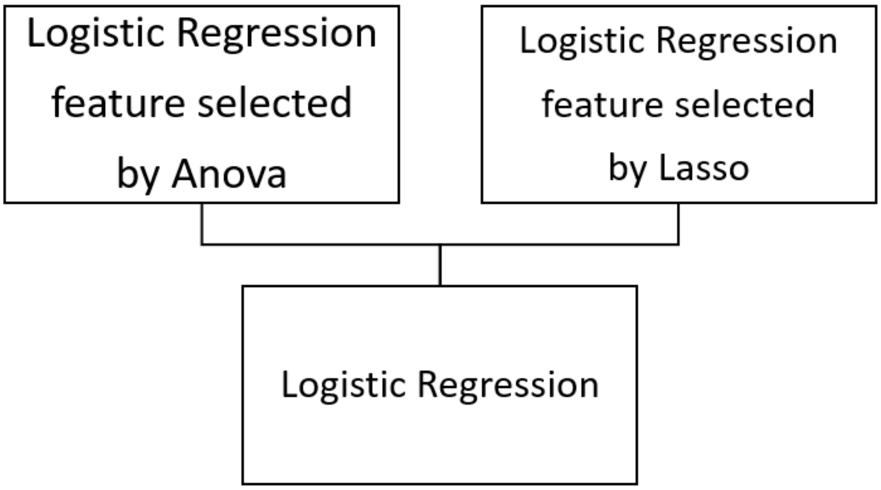
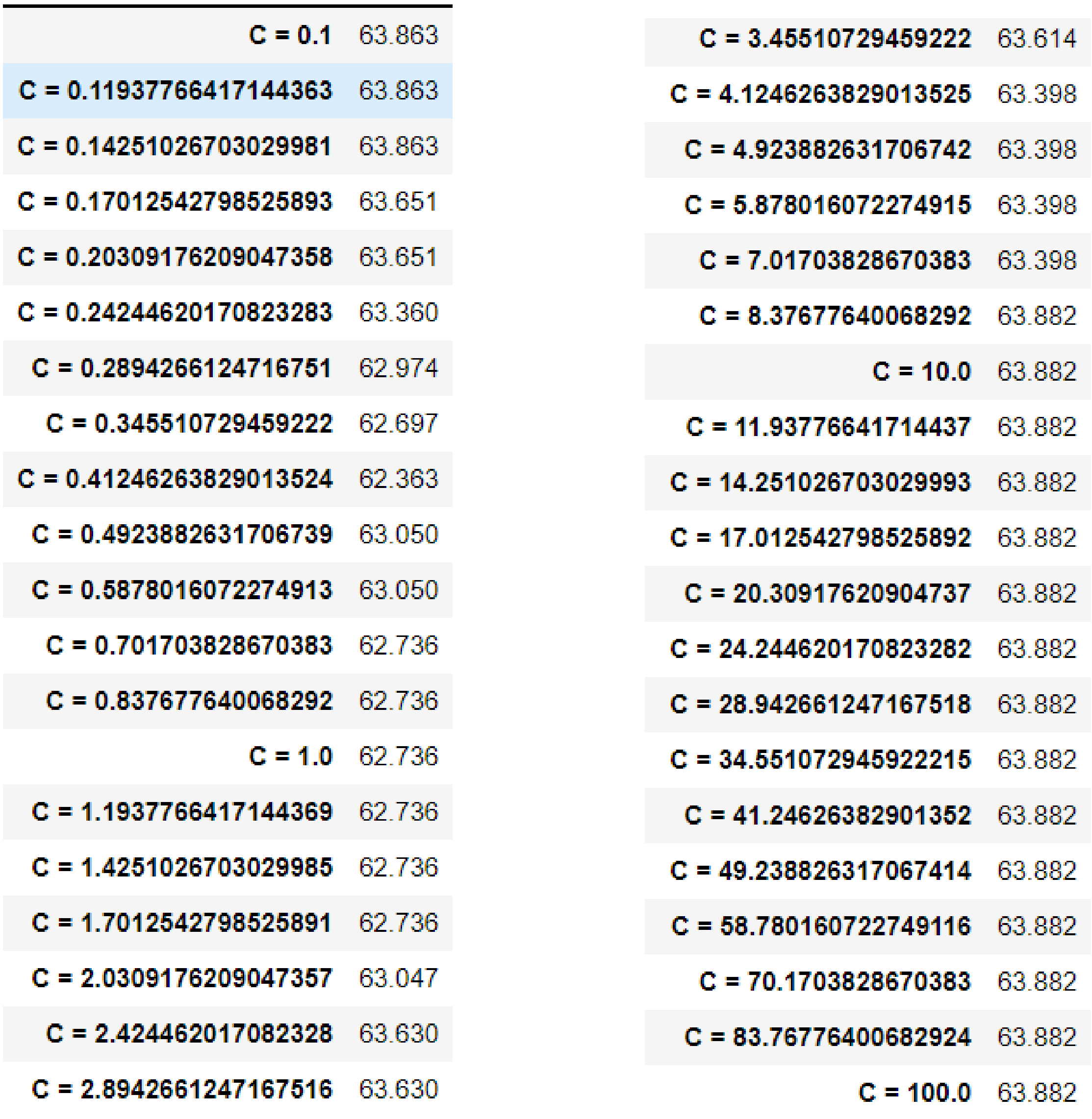

Logistic Regression feature selected by Anova
Logistic Regression feature selected by Lasso
Logistic Regression

Figure 11: The pipeline of the stacked modedel

The hyperparameters of the model in level1 is determined by the previous steps. Therefore, the hyperparameters we need to train are only the weights of L^1 penalty for the logistic regression in level2. Then I make the hyperparameters space as the followings:

$$C = 10^{-1}, 10^{-1+\frac{3}{4}}, ..., 10^{2-\frac{3}{4}}, 10^2$$

Again, to find the best hyperparameters in the above space we assume, we need to do a repeated cross validation, which splits the training data in 5-folds and repeat them in 100 iterations. From the sheet below, we can see that 63.882 is the highest average cross validation score in this 100 iterations, so we set our hyperparamters $C = 10, 10^{1+\frac{3}{4}}, ..., 10^2$ and get the same result. Without loss of generality, I set $C = 10$.



C = 0.1	63.863
C = 0.11937766417144363	63.863
C = 0.14251026703029981	63.863
C = 0.17012542798525893	63.651
C = 0.20309176209047358	63.651
C = 0.24244620170823283	63.360
C = 0.2894266124716751	62.974
C = 0.345510729459222	62.697
C = 0.41246263829013524	62.363
C = 0.4923882631706739	63.050
C = 0.5878016072274913	63.050
C = 0.701703828670383	62.736
C = 0.837677640068292	62.736
C = 1.0	62.736
C = 1.1937766417144369	62.736
C = 1.4251026703029985	62.736
C = 1.7012542798525891	62.736
C = 2.0309176209047357	63.047
C = 2.424462017082328	63.630
C = 2.8942661247167516	63.630
C = 3.45510729459222	63.614
C = 4.1246263829013525	63.398
C = 4.923882631706742	63.398
C = 5.878016072274915	63.398
C = 7.01703828670383	63.398
C = 8.37677640068292	63.882
C = 10.0	63.882
C = 11.93776641714437	63.882
C = 14.251026703029993	63.882
C = 17.012542798525892	63.882
C = 20.30917620904737	63.882
C = 24.244620170823282	63.882
C = 28.942661247167518	63.882
C = 34.551072945922215	63.882
C = 41.24626382901352	63.882
C = 49.238826317067414	63.882
C = 58.780160722749116	63.882
C = 70.1703828670383	63.882
C = 83.7677640068292	63.882
C = 100.0	63.882

Figure 12: The sheet of hyperparamters C

Next, we fit the model by whole training data and predict whether the patients in test data are sensitive or resistant to this drug. From the figure 14, result of ROC curve, we can see that the accuracy of the the model is around 0.706, and the f_β score is around 0.417.

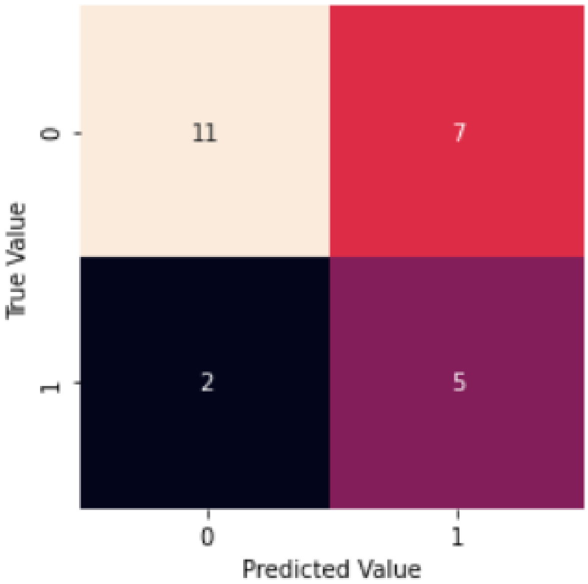


Figure 13: The confusion matrix of the stacked model with the data constructed by ClusterCentroids

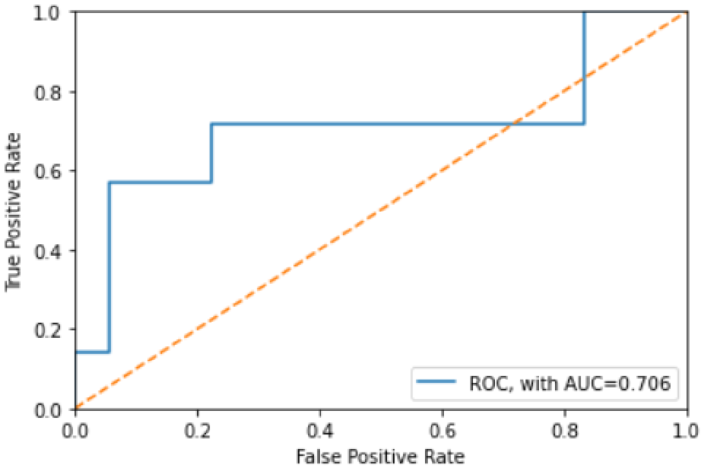


Figure 14: The ROC curve of the stacked model with the data constructed by ClusterCentroids

3 Conclusion

The result shows that Lasso get the highest accuracy and f_β score, then the stacked, and then Anova, which is different from my assumption. I thought that the stacked model has the highest accuracy and f_β score, since it combines the answers from the previous two model. However, it seems can not distinguish which answer is correct. The reason may be that I only choose two models in level1 such that the logistic regression in level2 have no enough to tell which one is correct by guessing. If there are more model in level1, then it can distinguish the true answer by the number of answers given by the previous model in level1. The other question is that the patients in the test data are different from the cell lines in the training data, so the prediction may not be right if our model is overfitting. Despite that problem, there is still a relation between the training data and test data. Moreover, the result above is much better than guessing whether the drug is sensitive to patients or not. Therefore, I thought the predictions of model has still a certain reference value.

4 Code

https://colab.research.google.com/drive/1W0ADm2EiqniHa-VBtf8XRoJ3a6hsajiI