Research on different resampling methods for imbalanced data

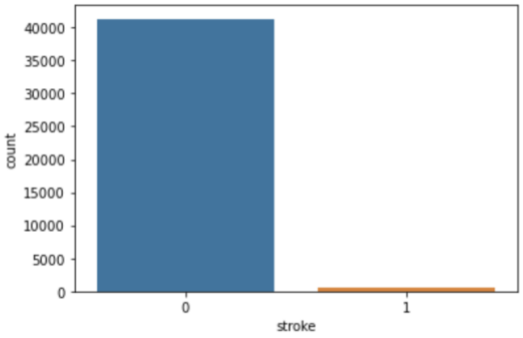
Introduction:

Classification is a technique classifying data into a given number of categories. In Machine Learning, most classification methods such as logistic regression, support vector machines, and random forest are applied to data with similar sample sizes for target variable. However, we will also encounter many data with different sample sizes for target variable. In the face of such unbalanced data, how to deal with it and how to predict the response variables with high precision by using classification method has aroused our deep concern. When dealing with unbalanced data, one of the most commonly used techniques is resampling, which includes two main types, undersampling and oversampling. Usually, people use undersampling when dealing with large datasets and oversampling when dealing with small datasets to get balance. As for our data, we speculate that the random undersampling method might work better because it has a huge amount of observations.

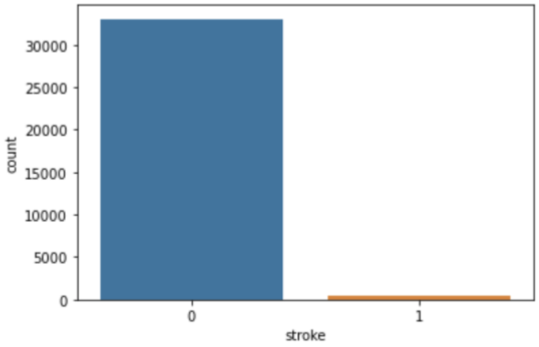
Description of the dataset:

- **Source**
  - https://www.kaggle.com/shashwatwork/cerebral-stroke-predictionimbalaced-dataset
  - Attribute Information: 1.Stroke 2.Age 3.Avg_glucose_level 4.bmi(Body mass index) 5.Heart_disease 6.Hypertension

- **Data processing**



To detect the incidence of stroke, we set non-stroke as 0 and stroke as 1. After processing the data by removing NAs from the original data, which NA will influence our conclusion, we have 41295 people who do not have a stroke and 643 patients with a stroke. From the plot, we can clearly see that there is a huge difference between the count of non-stroke and stroke. Therefore, We know clearly that our data set is in serious imbalance.



We split our original dataset into two parts, one of which contains random 80% of the observations training data and the other one of which contains the rest of 20% of the observations as test data. For training data, we have 33039 people who do not have a stroke and 511 patients with a stroke. The remained data is test data.

Methodologies:

I. **Resampling Technique**

In order to solve the unbalanced data, the resampling technique is to get new training data by balancing the number of minority classes and majority classes of data to reduce the influence caused by the unbalanced number of category data. The resampling methods we often use are under-sampling and oversampling. In the under-sampling, we use a simpler and more effective random sampling method in which we keep n observations from the minority class while removing N-n observations from the majority class with replacement. In the over-sampling, we use a random over-sampling and Synthetic Minority Oversampling Technique. The random over-sampling is that keeping N observations from the majority class, and then we randomly duplicate observations in the minority class as N number; The Synthetic Minority Oversampling Technique (SMOTE) is which we keep N observations from the majority class, and then we use SMOTE to generate a new minority class with the same number of observations as training data.

II. Algorithm

1. k-nearest neighbors algorithm (k-NN):

```
Step 1: For the Kth part, fit the model to the other K-1 parts of the data
Step 2: Calculated the mean squared error (MSE) of the fitted model when predicting the Kth part of the data as follow formula: MSE=$\frac{1}{n}\sum_{i=1}^{n} \quad (y_i - \hat{y_i})^2$
Step 3: We set K=1 to n and combine the n estimates of MSE and then we will average the all MSE as follow formula: CV(K)=$\frac{1}{K}\sum_{i=1}^{k} \quad MSE_i$
Step 4: Select which K has the smallest MSE.
End
Return K
```

2. SMOTE:

```
Step 1: Setting the minority class data set X, for each x ∈ X, and setting P=$\frac{the\ number\ of\ majority\ class}{the\ number\ of\ minority\ class} - 1$
Step 2: Finding the K nearest neighbors points of each x in minority class
Step 3: for i in 1:P do
    Randomly select one of the K nearest neighbors points and get new data set $X_k$
    Generate a random value r which is uniform distribution [0,1]
    Generate new data as follow formula: $X_i = x + r * |x - x_k|$ which $x_k \in X_k$
End
Return $X_1$,....,$X_p$
```

3. Classification algorithm:

We use the resampling method and then get the balance data, so we can use some classification methods for the new data. We prepare to use logistic regression, k-nearest neighbors algorithm, linear discriminant analysis, and random forest.

4. Accuracy score, Precision score, Recall score:

Using the classification algorithms, we will calculate the predicted value, and then we will compare the predicted value and actual value, finally, we will get the follow confusion table:

TN: Number of correctly mapped healthy cases=Number of (Y=0,$\widehat{Y} = 0$)	FP：Number of cases of patients who were wrongly mapped=Number of (Y=1, $\hat{Y} = 0$)
FN：Number of healthy cases that were wrongly mapped=Number of (Y=0, $\hat{Y} = 1$)	TP：Number of cases of correctly mapped patientsNumber of (Y=1,$\widehat{Y} = 1$)

Accuracy score = $\frac{TP+TN}{TP+FP+FN+TN}$ % which is used to reflect how close the measured value is to the known value.
Precision score = $\frac{TP}{TP+FP}$% which is used to reflect the reproducibility of measured values, even if they are far from the known values.
Recall score = $\frac{TP}{TP+FN}$ % which is how many true positives are found, that is, how many correct hits are also found.

Note: Different confusion table will be constructed for different classification algorithms to show the performance of the models.

Implementation Details:

- **No resampling (Original):** We use the training data set without any modification.
- **Random Under-Sampling**: We keep 511 observations from the minority class and remove 40652 observations from the majority class with replacement. Then, we get the new balance training data, which each class's size is 511. (imblearn.under_sampling.RandomUnderSampler)
- **Random Over-Sampling**: We keep 33039 observations in the majority class and randomly duplicate observations in the minority class as 33039, so that we get the new balance training data, which each class is a 33039 number. (imblearn.over_sampling.RandomOverSampler)
- **Synthetic Minority Oversampling Technique (SMOTE)**: We keep 33039 observations from the majority class, and then we use SMOTE to generate a new minority class with the same number of observations as training data. Then, we get a balanced data set, and each class has 33039 observations. We set the number of nearest neighbors as K=1 for the over-sampling process. (imblearn.over_sampling.SMOTE)

We utilize four various classification algorithms in python to test the performance of each resampling method:

- **logistic regression** (sklearn.linear_model.LogisticRegression)
- **k-nearest neighbors algorithm** (sklearn.neighbors.KNeighborsClassifier)
- **linear discriminant analysis** (sklearn.discriminant_analysis.LinearDiscriminantAnalysis)
- **random forest** (sklearn.ensemble.RandomForestClassifier)

Also, we select the best k for the SMOTE and KNN algorithm. We use cross-validation to find the best K, in addition, we use GridSerachCV() to fit the model because it is slow but may have better results. Then we choose cv=10 because the higher the cross-validation, the lower the error.

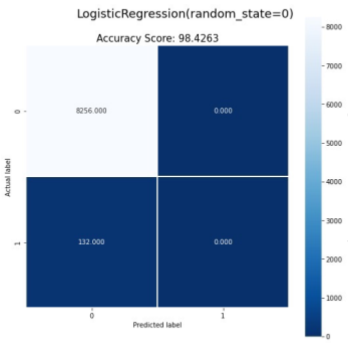
Conclusion:

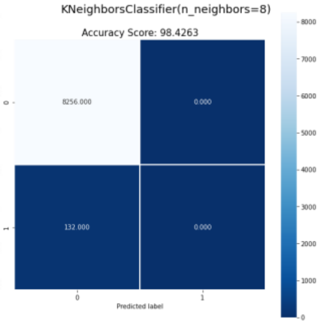
From the data analysis, through the three test methods including precision score, recall score, and accuracy score, as for the dataset we are processing, Random Under-Sampling, Random Over-Sampling and Synthetic Minority Oversampling Technique (SMOTE) works fine. Even though the accuracy score of random under-sampling shows a lower accuracy score than the two over-sampling tests, it has better performance in the precision score and the recall score. Meanwhile, the knn method may lead to poor performance when operating Random Over-Sampling and Synthetic Minority Oversampling Technique (SMOTE). Therefore, we conclude that overall for a dataset with a large sample size like the dataset of stroke, we prefer to use a random under-sampling method.

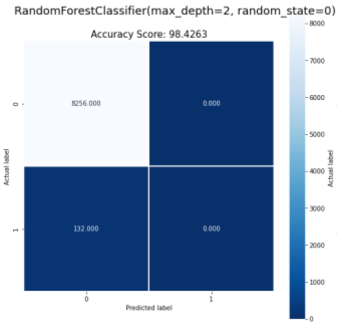
During the research, we found some advantages and disadvantages of Under-sampling as well as Advantages and disadvantages of oversampling. The advantage of under-sampling is to reduce run time because we are deleting huge amounts of data when the training dataset is big; however, the disadvantage of under-sampling is that the deleted data might be significant that may have a huge influence on the result, and the data left could also lead to a biased result. The advantage of oversampling is that it keeps all the data, which means that the result would be based on all the information without bias. Additionally, overfitting possibility increase is the main disadvantage of over-sampling because it replicates the observations from the minority class.

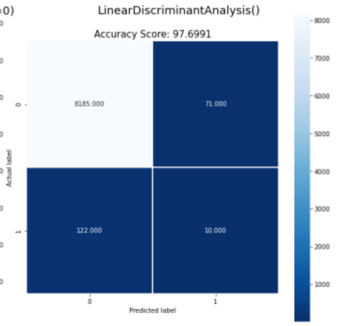
Other Results and Interpretation:

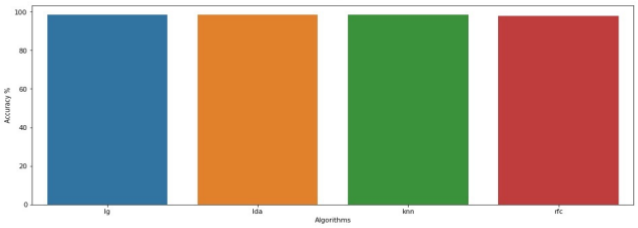
- Original data:

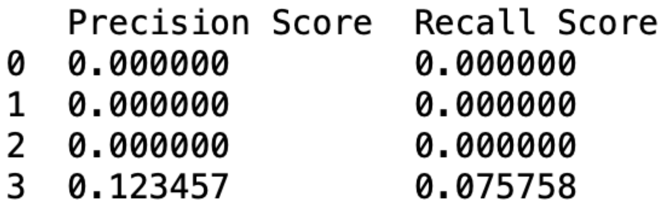








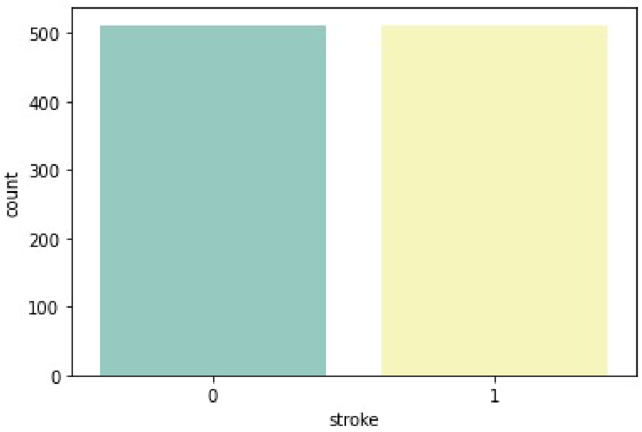




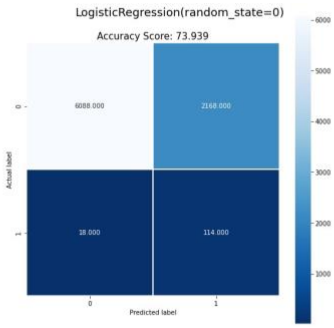
```
   Precision Score  Recall Score
0  0.000000         0.000000
1  0.000000         0.000000
2  0.000000         0.000000
3  0.123457         0.075758
```

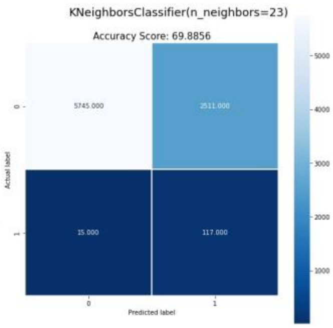
Before applying methods of resampling, we operate four kinds of algorithms of model building including logistic regression, k-nearest neighbors, linear discriminant analysis, and random forest to the original data. Although the accuracy scores of these models are extremely high, the precision scores and recall scores for logistic regression, k-nearest neighbors, and random forest are zero, which means that the model does not predict any case due to imbalanced data. Additionally, Linear discriminant analysis models do some correct predictions, but it is not enough at all. We again prove that the imbalanced data will influence the result badly. Therefore, we need to apply a resampling method to improve the dataset.

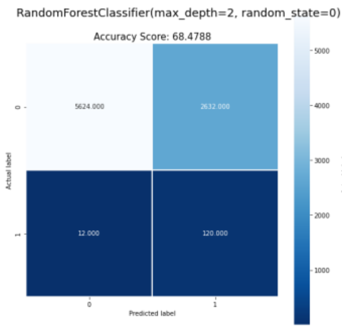
- Random Under-Sampling:

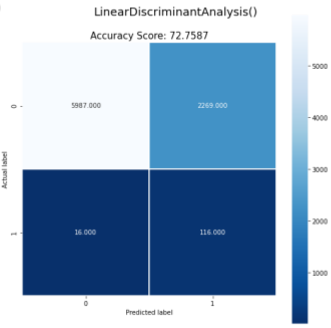


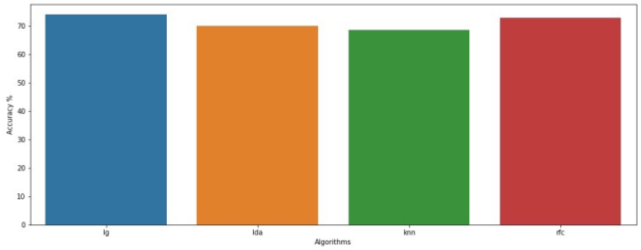
With random undersampling, we preserve 511 observations from the minority class of stroke patients and remove data from the majority class of stroke patients until we have the same number of observations as the minority class, which is 511 observations. Then, the new balance training data, which each class's size is 511, will be used to build our models. The left figure is the histogram of the new training dataset.

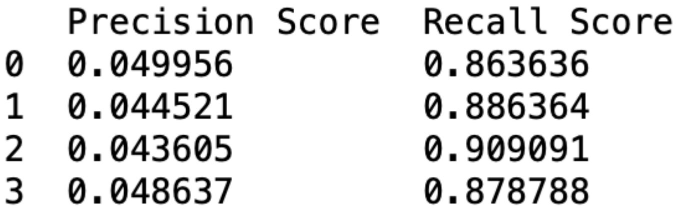








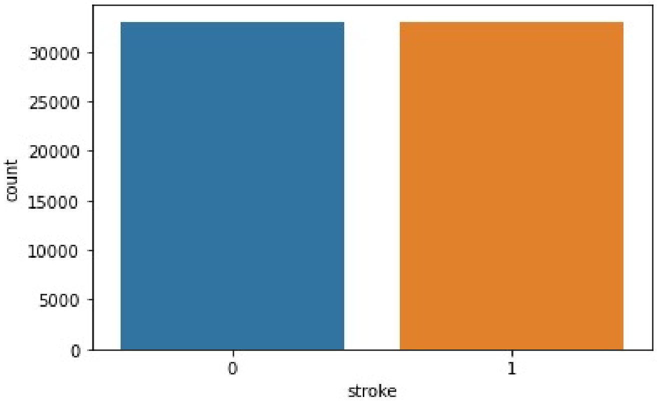




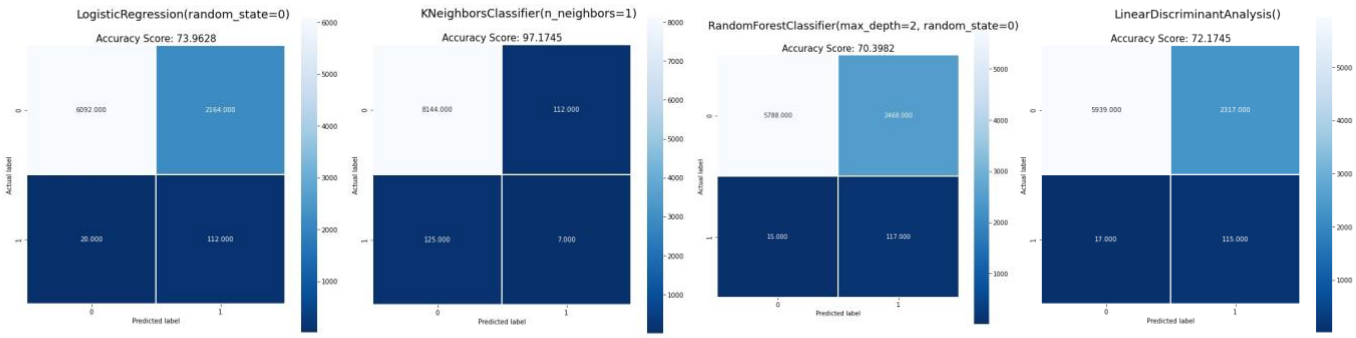
```
   Precision Score   Recall Score
0  0.049956          0.863636
1  0.044521          0.886364
2  0.043605          0.909091
3  0.048637          0.878788
```

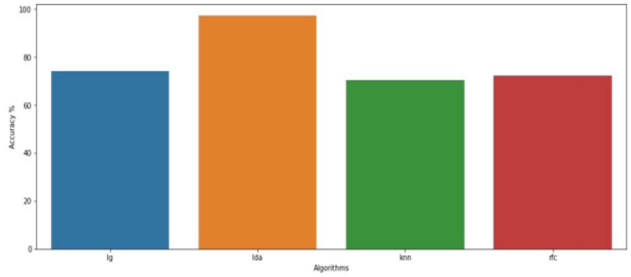
Through the operation of four algorithms of model building including logistic regression, k-nearest neighbors, linear discriminant analysis and random forest, we found that the accuracy scores of these four algorithms get lower than the original data. The average accuracy score is around 70%, which is normal. However, the precision scores and recall scores improve a lot compared to the original data. The models made some correct predictions and found most of the stroke patients.

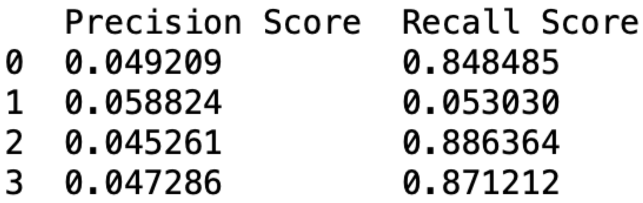
- Random oversampling:



With random oversampling, we keep 33039 observations in the majority class for the people who do not have stroke and randomly duplicate observations in the minority class of the stroke patients until it reaches 33039 observations. And we will use the new balanced dataset as our training data. The histogram of the training data is presented in the left side.



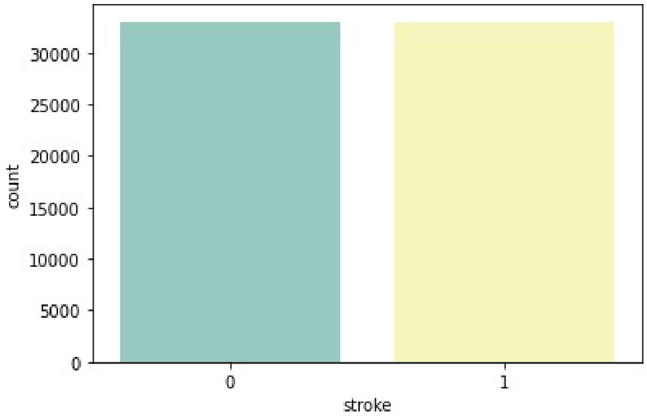




```
   Precision Score  Recall Score
0  0.049209         0.848485
1  0.058824         0.053030
2  0.045261         0.886364
3  0.047286         0.871212
```

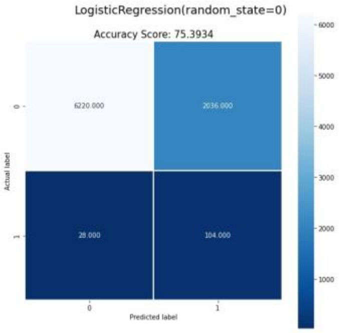
Again, we use four modeling algorithms, including logical regression, K-nearest neighbor, linear discriminant analysis, and random forest. We can observe that the accuracy of these models has increased to a certain extent, especially the accuracy score of knn model has increased to a very high value, which may be because knn model didn't make too many predictions about stroke at all. Furthermore, the recall score of knn method becomes extremely low, which shows that there are very few stroke patients in knn method. Therefore, we think that in knn model, most stroke patients are considered non-stroke patients.

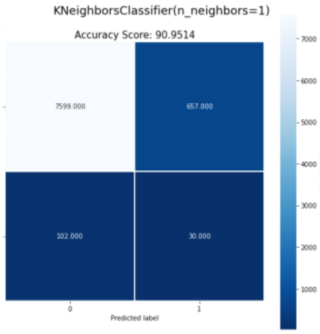
However, the precision scores and recall scores of other methods show high accuracy. Therefore, random oversampling improved our data model, but did not improve knn model.

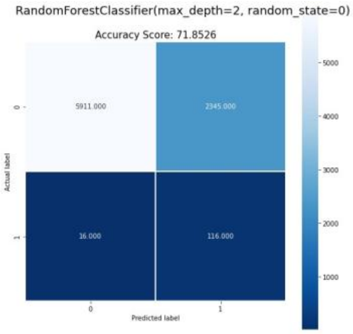


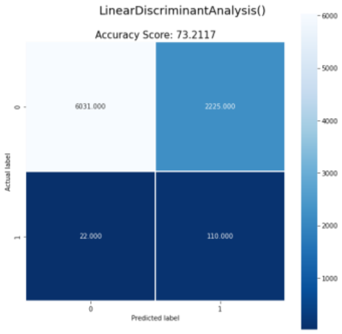
- Synthetic Minority Oversampling Technique (SMOTE):

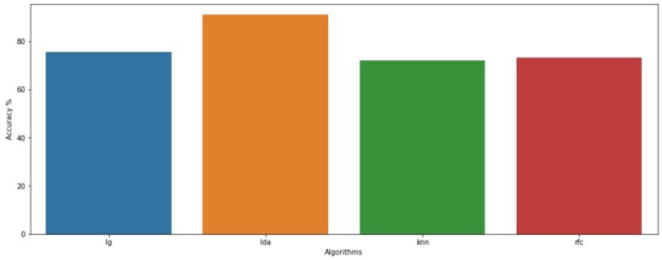
We keep 33039 observations from the majority class of people who do not have strokes, and generate a new minority class of stroke patients that has the same number of observations as the majority class with the method of SMOTE. The left side show the histogram of training data.

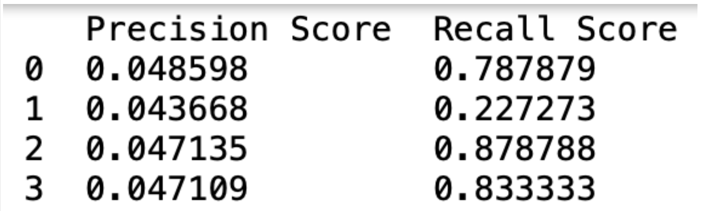












```
   Precision Score  Recall Score
0  0.048598         0.787879
1  0.043668         0.227273
2  0.047135         0.878788
3  0.047109         0.833333
```

This time operating the different methods, the results of three different scores (accuracy score, precision score, recall score) of these models look similar to the results of a random oversampling method. Overall, there is only a slight difference between the scores of the SMOTE method and the random oversampling method. Also, the accuracy score of knn model is pretty high but the low recall score shows the insufficiency of the knn model clearly.