CS5014- P1 report

Part1 : Data Processing

(a) How did you load and clean the data, and why?

Firstly, I changed crx.data into crx.csv, while adding the column names which is the feature name(A1, A2,...,A16). Then I discovered that there was 37 values(5%) missing. Because missing values accounted for a small proportion of the entire data set, I directly removed samples with missing data. I know that data is precious, and next time I will use random forest to complete the default value.

(b) How did you split the data into test/train set and why?

First of all, I set the test_size = 0.5, which means 50% is the train set and 50% is the test set. I found that the results of model fitting are not very good. Then I tried to divide the training set into 20%~80% to observe the prediction accuracy. I found that as the number of training sets increases, the accuracy of the model will gradually increase.

Then I used a random division strategy when dividing the training set. That is, both times are divided by 50%, and the training set obtained is different. Based on the above results, I drew a score fluctuation curve based on the number of training sets, shown in the Figure 1.

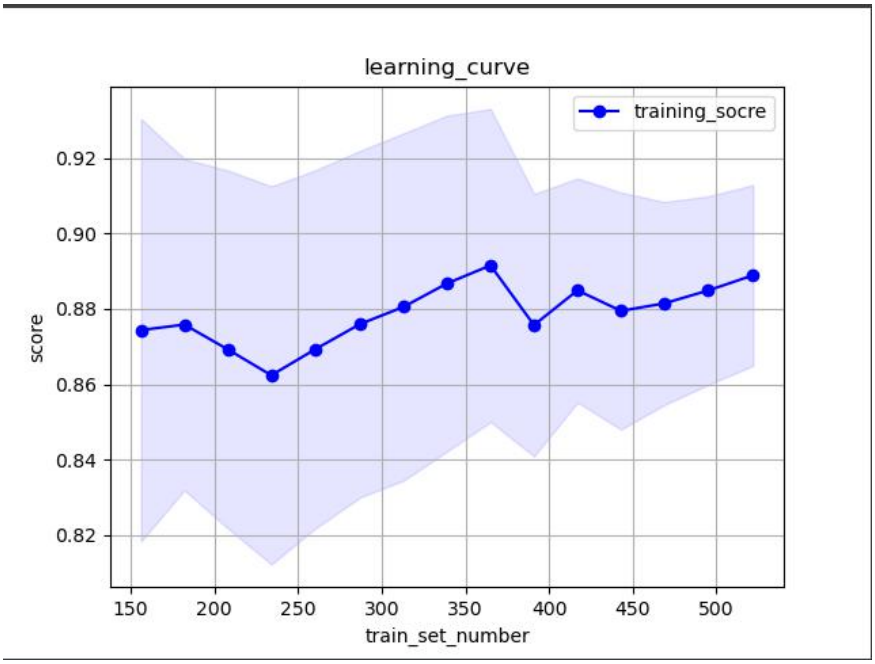


Figure1 Score Fluctuation Curve

The X axis represents the number of training sets, the Y axis represents the score, and the blue area represents the fluctuation of the score under random allocation.

(c) How did you process the data including encoding, conversion, and scaling, and why?

For the conversion, because the logistic regression model only accepts numeric values, I use regularization to convert the features of the object type to numeric types, converting 15 attributes from A1 to A15 into 46 attributes.

Foring scaling, according to the observation of the features (Figure 2), the data fluctuation range of A15 and A14 is shortened, which will affect the speed of model convergence. So we should scale the data. I did not process data in this direction in this assignment.

```
In [230]: new_df.describe()
```

Out[230]:

	Unnamed: 0	A2	A3	A8	A11	A14	A15
count	653.000000	653.000000	653.000000	653.000000	653.000000	653.000000	653.000000
mean	326.000000	31.503813	4.829533	2.244296	2.502297	180.359877	1013.761103
std	188.649145	11.838267	5.027077	3.371120	4.968497	168.296811	5253.278504
min	0.000000	13.750000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	163.000000	22.580000	1.040000	0.165000	0.000000	73.000000	0.000000
50%	326.000000	28.420000	2.835000	1.000000	0.000000	160.000000	5.000000
75%	489.000000	38.250000	7.500000	2.625000	3.000000	272.000000	400.000000
max	652.000000	76.750000	28.000000	28.500000	67.000000	2000.000000	100000.000000

Figure 2 Features Describe

(d) How did you ensure that there is no data leakage?

When the data you are using to train a machine learning algorithm happens to have the information you are trying to predict(Brownlee, 2016). Because there is no specific feature name given to the attribute in the data set of this job, we can't filter out or deal with this attribute in the first place. After analyzing the parameter results (Figure 3). A6 attribute value ‘A6_ff’ or ‘A6_cc’ can greatly increase or decrease the direction of classification. Therefore, in the subsequent classification, we should try not to consider the ‘A6_ff’ and ‘A6_cc’ attributes, or use some methods to reduce the influence of this attribute. In conclusion, in this prediction model, I think there is a certain amount of data leakage.

14	A6_aa	[0.0]
15	A6_c	[-0.24524009706851202]
16	A6_cc	[1.3907473176708556]
17	A6_d	[0.0]
18	A6_e	[0.7197672130150972]
19	A6_ff	[-1.3534122207697308]
20	A6_i	[-0.014926405513986999]
21	A6_j	[0.0]
22	A6_k	[-0.9377101030143529]
23	A6_m	[0.0]
24	A6_q	[0.3814844346328295]
25	A6_r	[0.0]
26	A6_w	[0.0]
27	A6_x	[0.49281924294209334]

Figure 3: Parameter Result of A6.

Part2:

(a) Train using penalty='none' and class_weight = None. What is the best and worst classification accuracy you can expect and why?

When I set the class_weight = None, the accuracy is 0.851, while the accuracy is 0.854 when I use the class_weight. There is no big difference between them. Because in this data set, the proportions of label 1(44.5%) and label 0(55.5%) are very close. We don't really need to use class_weight to reset the weight.

(b) Explain each of the following parameters used by LogisticRegression in your own words : penalty, tol, max_iter.

penalty: The regularization of the loss function is to prevent the occurrence of underfitting and overfitting problems. That is, add a penalty term after the loss function. I understand that penalty is similar to setting a transformable step. And in the gradient descent method, the parameters can be iterated better and faster in this step. Also, in penalty, we can choose different loss function convergence methods, such as gradient descent (liblinear), Quasi-Newton Method (lbfgs) and so on. If the loss function is not continuous and differentiable, we can choose l1.

tol: The conditions that when the algorithm stops. That is, stop the iteration when the parameter change value is less than tol.

max_iter: If the function never converges, that is, it is less than tol. Then it stops after iterating max_iter times.

(c) Train using balanced class weights (setting class_weight='balanced'). What does this do and why is it useful?

class_weight: If class_weight is balanced, the class library will calculate the weight based on the training sample size. The larger the sample size of a

certain type, the lower the weight, and the smaller the sample size, the higher the weight. In this data set, because the difference between the total number of labels of ‘0’ and ‘1’ is not large, the balance setting has not been greatly improved. However, when there are significant differences in the amount of different types of samples in the data set, setting the balanced weight will prevent data leakage.

(d) LogisticRegression provides three functions to obtain classification results. Explain the relationship between the vectors returned by predict(), decision_function(), and predict_proba().

predict(): This prediction is the final label, which is 1 or 0. That is, classify those with a confidence level greater than 0.5 as class 1, and classify those with a confidence level of less than 0.5 as class 0.

decision_function(): The confidence score for a sample is proportional to the signed distance of that sample to the hyperplane(scikit-learn.org, n.d.). For example, when there is only one attribute, confidence score is the distance from the test set data point to the classification line. So, in linear regression, the smaller the value, the better. In logistic regression, the opposite is better.

predict_proba(): Because logistic regression uses maximum likelihood estimation, samples that are classified into class 1 at the same time contain different probabilities. That is, the greater the probability, the greater the confidence score. The farther this sample point is from the hyperplane. The higher the confidence that this sample is classified into this category.

Part3 : Evalution

What is the classification accuracy of your model and how is it calculated? Give the formula.

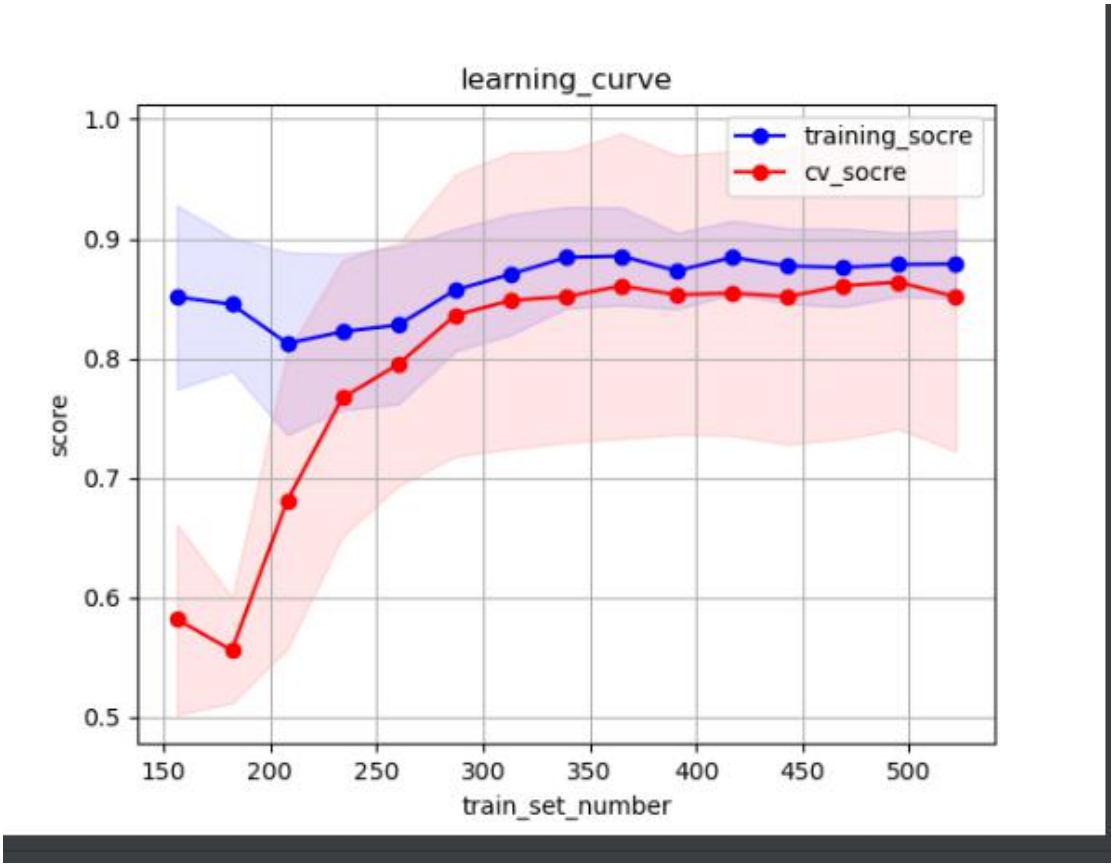
Classification accuracy is the evaluation of the classification results on the test set.

classification accuracy = 1 - Bad_Case / Test_Set

(b) What is the balanced accuracy of your model and how is it calculated? Give the formula.

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

(d) Plot the precision-recall curve and report the Average Precision (AP) for your algorithm.What is th erelationship between AP and the PR curve?



Finally, cross-validation is used to test the over-fitting and under-fitting of the model.

Both the training set and the validation set score high, indicating that the model is not under-fitting. In addition, as the number of samples increases, the scores of the training set and the validation set are getting closer and closer, and there is no big gap between the two. Explain that the model is not over-fitting (over-fitting will result in high scores on the training set and low scores on the validation set.