CS5014 P1 Report

Part 1: Data Processing

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

A: I use pandas to read the data in as a DataFrame and then start pre-processing the data. Steps include missing value filling/deleting, outlier handling, one-hot encoding, normalisation and splitting.
This is because missing data affects the integrity of the data set and outliers make the data unsmooth and not easy to fit. Standardisation reduces the effect of features with large variance, allowing the gradient to converge faster. And one-hot coding allows the classification attributes to be represented numerically and avoids ordered numbers affecting the classification results.

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

A:I used the train_test_split() method in Sklearn.model_selection to divide 70% of the dataset into a training set and the rest as a test set. Since it is not reasonable to use the model's fit to the training set to represent its performance, the test set is used to test its generalization.

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

A:In the missing value filling, I deleted records with missing value because they represent a relatively small proportion of the data (5%) and in the test, the model is slightly more accurate than filling if these records are removed.
When dealing with outliers, I use Tukey's Test to find the outliers and replace them with the median since the attributes do not fit the normal distribution.
In order to translate the classification attributes into the model I should use one hot encoding. The encoding is done by converting True/False to 0/1 and then using Pandas.get_dummies()
Standardisation reduces the variation between features of different orders of magnitude and improves the accuracy of the model. So I use

sklearn.Standscaler() to specify that all continuity features take values from -1 to 1 and are normally distributed.

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

A:Plot the Pearson correlation coefficient matrix after normalising the data and check if there are features that are extremely highly correlated with the target feature and if so consider not using the feature to prevent data leakage.

Part 2: Training

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

The worst classification accuracy should be around 50%, as the category distribution is 44.5%/55.5%, even if the classifier predicts all outputs to be POSITIVE, then the correct rate is still around 50%.

The best result is that the model fits the dataset well, so the accuracy should be above 90%

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

Penalty: Classes of regularization formula to prevent overfitting, respectively L1, L2, elasticnet or none. Some sovler can only use L2 regularization.

Tol:Lower limit of the cost function. The classifier stops iterating when the gradient converges to its specified magnitude.

Max_iter: Maximum number of iterations.The classifier is forced to end when it reaches a specified number of iterations.

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

The classifier will assign weights based on the training sample size. If a class has a large sample size then the weights are lower and vice versa.
The balanced class weight is useful, especially when the dataset is unbalanced, as it allows classes with fewer samples to be weighted higher and more samples to be classified in that category. This makes the classifier's predictions more accurate(Brodersen et al., 2010).

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

The vector of predict() is the result of which category the sample belongs to
The vector of decision_function() is the confidence that the sample is positive
The matrix of predict_proba() outputs the probability that the sample is positive and negative.
In contrast to predict_proba(), the decision_function() only considers the possibility of a positive sample, but they express the same classification result, and both of these can inferred the predict() output

Part 3: Evaluation

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

Accuracy is the proportion of the total sample that is correctly predicted(*Precision and recall - Wikipedia*, 2020).
The formula is :

$$\frac{TP+TN}{TP+TN+FP+FN}$$

True Positive（TP）: the number of positive samples predicted as positive samples

False Positive（FP）: the number of negative samples predicted as positive samples

False Negative（FN）: the number of positive samples predicted as negative samples

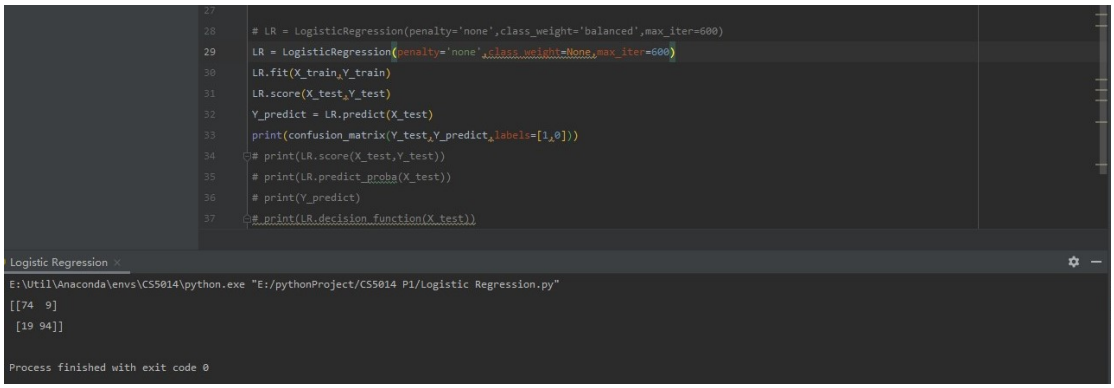
True Negative（TN）: the number of negative samples predicted as negative samples

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

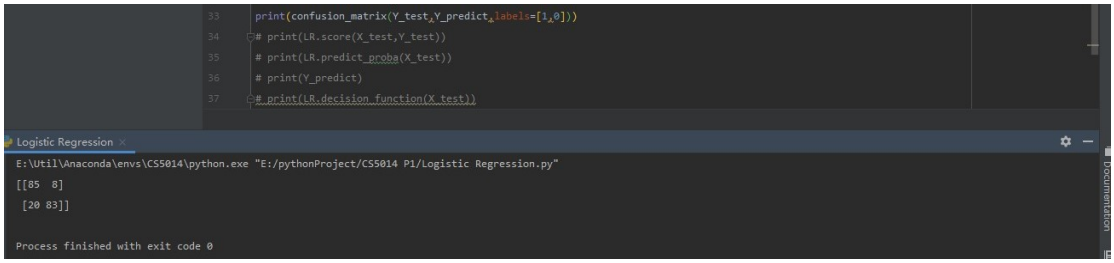
Balanced accuracy is a better metric to use with imbalanced data. It accounts for both the positive and negative classes by calculating the percentage of true positive samples in all the positive samples and the that of true negative samples in all the negative samples(*Precision and recall - Wikipedia*, 2020). The formula is:

$$\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP} \right) \times \frac{1}{2}$$

(c) Show the confusion matrix for your classifier for both unbalanced (2a) and balanced (2b) cases. Discuss any differences.



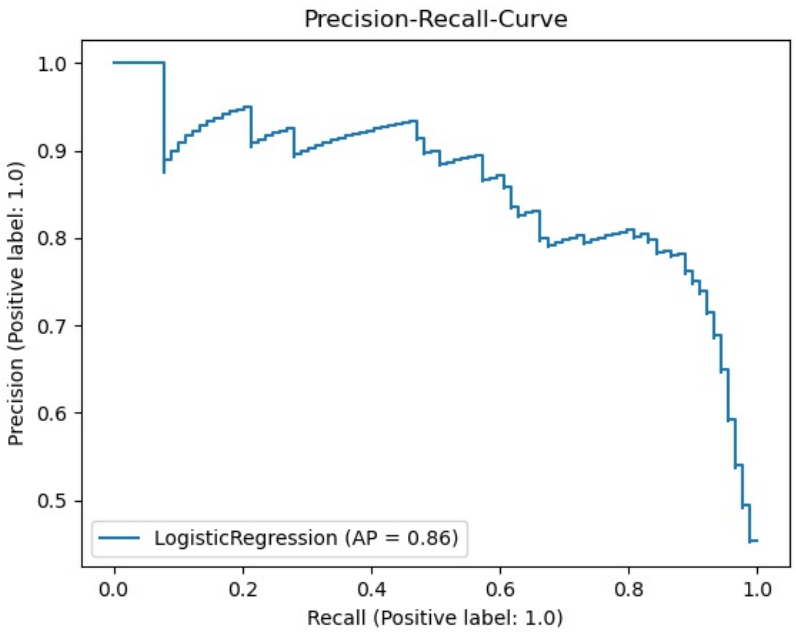
Confusion matrix with unbalanced class weight



Confusion matrix with balanced class weight

The number of positive samples for the whole dataset is lower (44.5% vs. 55.5%). And the number of positive samples predicted by the classifier is lower when compared to the case where a balanced class_weight is set.
So with balanced set the positives have more weight and the classifier tends to predict the samples as positive.However this data set is relatively balanced, so this trend is not apparent here, and balanced class weight would be much helpful in imbalanced dataset.

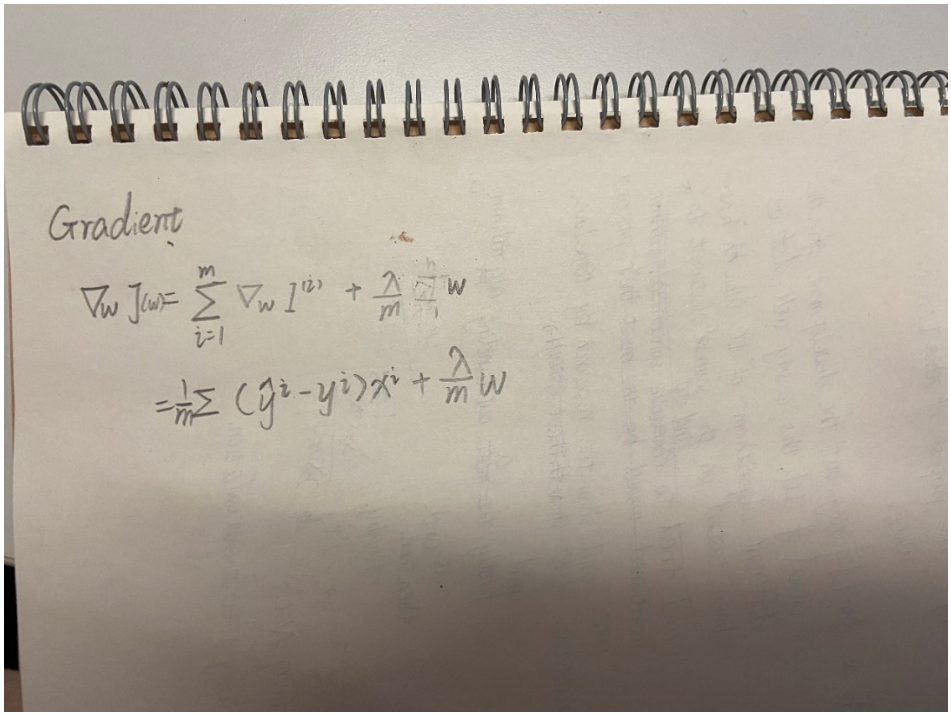
(d) Plot the precision-recall curve and report the Average Precision (AP) for your algorithm. What is the relationship between AP and the PR curve?

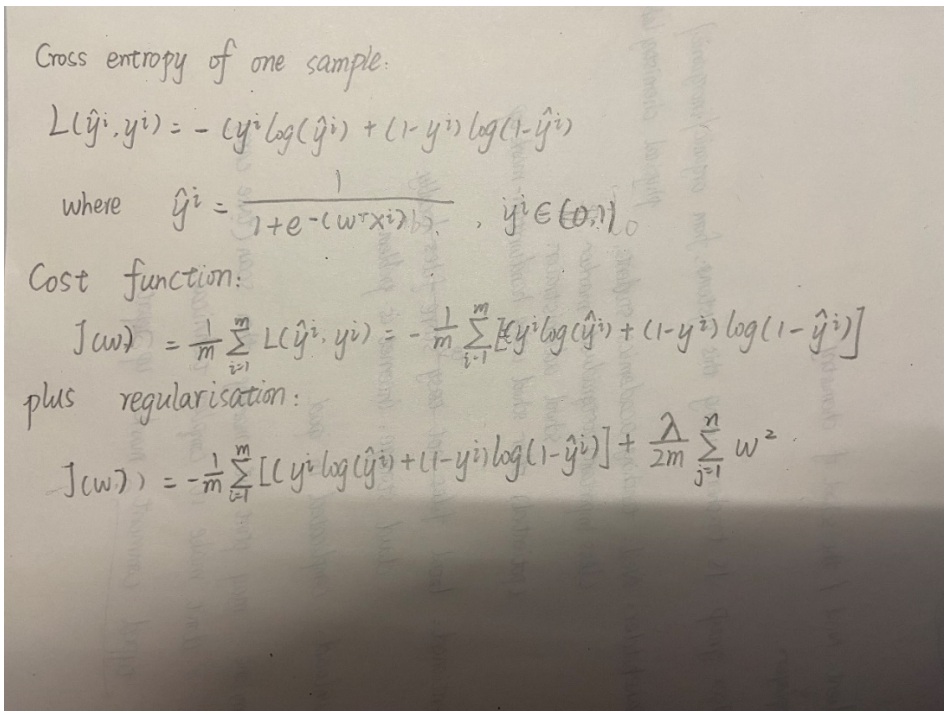


The average precision is the mean of a category Precision, which is the integral of the precision-recall curve(*Information retrieval - Wikipedia*, 2020). The number of it is 0.85.

Part 4：

(a) Set the penalty parameter in LogisticRegression to ‘l2’. Give the equation of the cost function used by LogisticRegression as the result. Derive the gradient of this l2- regularised cost.





(c) Compare the results of regularised and unregularised classifiers on the expanded data and explain any differences.

The use of regularisation prevents overfitting however no significant improvement in the accuracy of the model occurs. It is possible that the model was not overfitted and so the regularisation was not significant

Reference

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010, 23-26 Aug. 2010). The Balanced Accuracy and Its Posterior Distribution. 2010 20th International Conference on Pattern Recognition,

Information retrieval - Wikipedia. (2020). Retrieved 3/5 from https://en.wikipedia.org/w/index.php?title=Information_retrieval&oldid=793358396#Average_precision

Precision and recall - Wikipedia. (2020). Retrieved 3/5 from https://en.wikipedia.org/wiki/Precision_and_recall