COMP9417: Homework Set #2

z5113817

University of New South Wales — July 17, 2021

Question 1

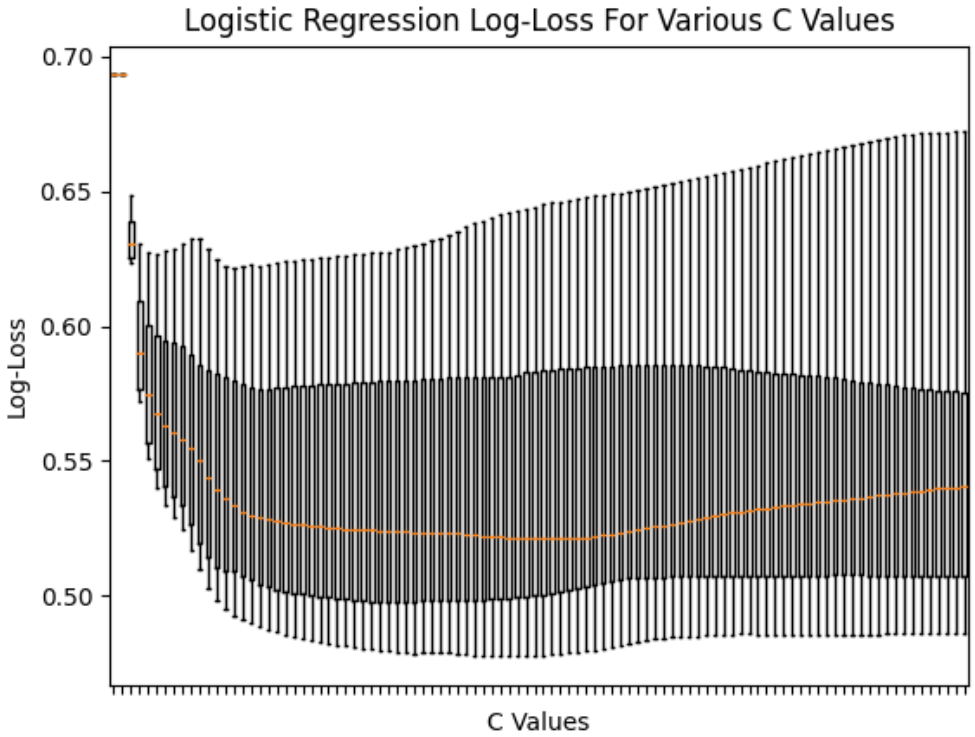
a

The possible values for both y_i and $\tilde{y}_i$ are binary. Even though they have different values ($y_i \in \{0, 1\}$ and $\tilde{y}_i \in \{-1, 1\}$), the objective of each logistic regression implementation is to divide the dataset into 2 classifications. Because of this, the actual value that each classification has will not affect the parameters that the regression is attempting to optimise ($(\hat{\beta}_0, \hat{\beta})$ and $(\hat{w}, \hat{c})$). Therefore the solutions for the parameters being minimised by each regression will be the same.

C is a hyper-parameter that adjusts the sensitivity that the model has to its coefficients. Compared with the standard LASSO parameter λ, C is a multiple of the Loss function whereas λ is a multiple of the Penalty.

b

Boxplot of testing accuracy for each value of C:



The value of C returning best results: **0.18794747474747472**
The testing accuracy of this model: **76%**

c

From GridSearchCV:

The value of C returning best results: **0.012219191919191918**

The testing accuracy of this model: **75.2%**

In our answer for b , we determined the "best" value of C as the value which corresponded to the average lowest *log-loss* value across all folds. The value from the *GridSearchCV* are different because, by default, it determines the "best" value of C as the one which corresponds to the average highest *score* across all folds [1].

We can modify the *GridSearchCV* class by providing our own metric for *scoring*. The following code is a scorer that uses the smallest *log-loss* value as its scoring metric.

```
scoring = make_scorer(
    log_loss,                # The sklearn implementation of log_loss
    greater_is_better=False, # A smaller log_loss is a better value
    needs_proba=True         # Calculating log_loss needs the probability predictions
)
```

This yields the following results:

*grid_lr.best_estimator_*: LogisticRegression(C=0.18188787878787877, penalty='l1', solver='liblinear')
*grid_lr.best_score_* (lowest log-loss): -0.5374067706086373

This value of *C* matches my value in b .

[1] See *scoring* parameter in documentation. If the estimator provided exposes a *score* method and a value for *scoring* is not provided, then *score* is used to determine the "best" value of C