Big Data Analytics Techniques and Applications

Homework 4

309709022 陳政廷

Q1: Show the predictive framework you designed.
Q2: Explain the validation method you use
Q3: Explain the evaluation metric you use.
Q4: Show the validation results and give a summary of results.

在此將整個預測流程分成 Data preprocessing、Modeling、Result

a. Data preprocessing

(1) 讀入資料

(2) 建立 Target Variable(Delay or not)

```
#########################
# 定義Target variable #
#########################
train_data = train_data.withColumn('Delay',
when((train_data.ArrDelay <0) & (train_data.DepDelay<0), 1).otherwise(0))
valid_data = valid_data.withColumn('Delay',
when((valid_data.ArrDelay <0) & (valid_data.DepDelay<0), 1).otherwise(0))
```

✓ 0.1s

(3) 缺失值填補(針對數值填入平均值、類別填入眾數)

(4) 將 validation data 出現的新 label 移除(因 training set 沒有對應 label 支持)

(5) 確認是否有類別不平衡的狀況

```
+-----+-------+
|Delay|  count|
+-----+-------+
|    1|4623786|
|    0|8994024|
+-----+-------+
```

看起來較還好兩者比例相差約一半，後續仍有做 Oversampling 呈現的結果，但

其整體表現較差

(6) 將類別欄位進行 one hot encoding 並使用 VectorAssembler 產出後續放入預測模型的 features

b. Modeling

本次使用 LogisticRegression 且參數設定如下

```
lr = LogisticRegression(maxIter=10^3,regParam=0.1,featuresCol="features",
labelCol="Delay")
lrmodel = lr.fit(train_data)
train_pred = lrmodel.transform(train_data)
valid_pred = lrmodel.transform(valid_data)
```

由於電腦關係若使用 Cross validation 將會產生記憶體不足的問題，故沒有使用

c. Result

分別選用 precision、recall、f1-score、accuracy 作為 eval_metrics 且結果如下

```
train_precision:  0.6789184943257511
train_recall:  0.9615665913277527
train_f1score:  0.5915148567029522
train_auc:  0.6742689169550757
valid_precision:  0.6792266763454395
valid_recall:  0.9630940122349099
valid_f1score:  0.583343397244667
valid_accuracy:  0.6723429552694955
```

TP、FP、TN、FN 的狀況

confusion	count
TP	4307252
TN	296022
FN	259026
FP	1990652

從中可以發現大致上沒有產生 overfitting 的問題，且模型的 recall 非常高，代表在事實為真(確實沒有發生 Delay 的狀況下)此模型能非常精準預測其結果，而從其他指標可以看出，我們最關心的 Delay 為真且預測為真的狀況，預測效果不是很好，故亦有嘗試使用 Oversampling 的資料進行建模，得出以下結果

```
train_precision:  0.9279557072862297
train_recall:  0.3322259313517509
train_f1score:  0.6133910193234708
train_auc:  0.6532269689262626
valid_precision:  0.8619001015110084
valid_recall:  0.3108978472182377
valid_f1score:  0.4878977598721202
valid_accuracy:  0.507643129559349
```

其中可以發現，雖然有改善 Delay 為真且預測為真的狀況，但在整理得 valida accuracy 下降非常多，故未來可以考慮以調參或是不一樣的資料前處理方式試著增加整體的準確度以及對於 Delay 為真且預測為真狀況的優化。