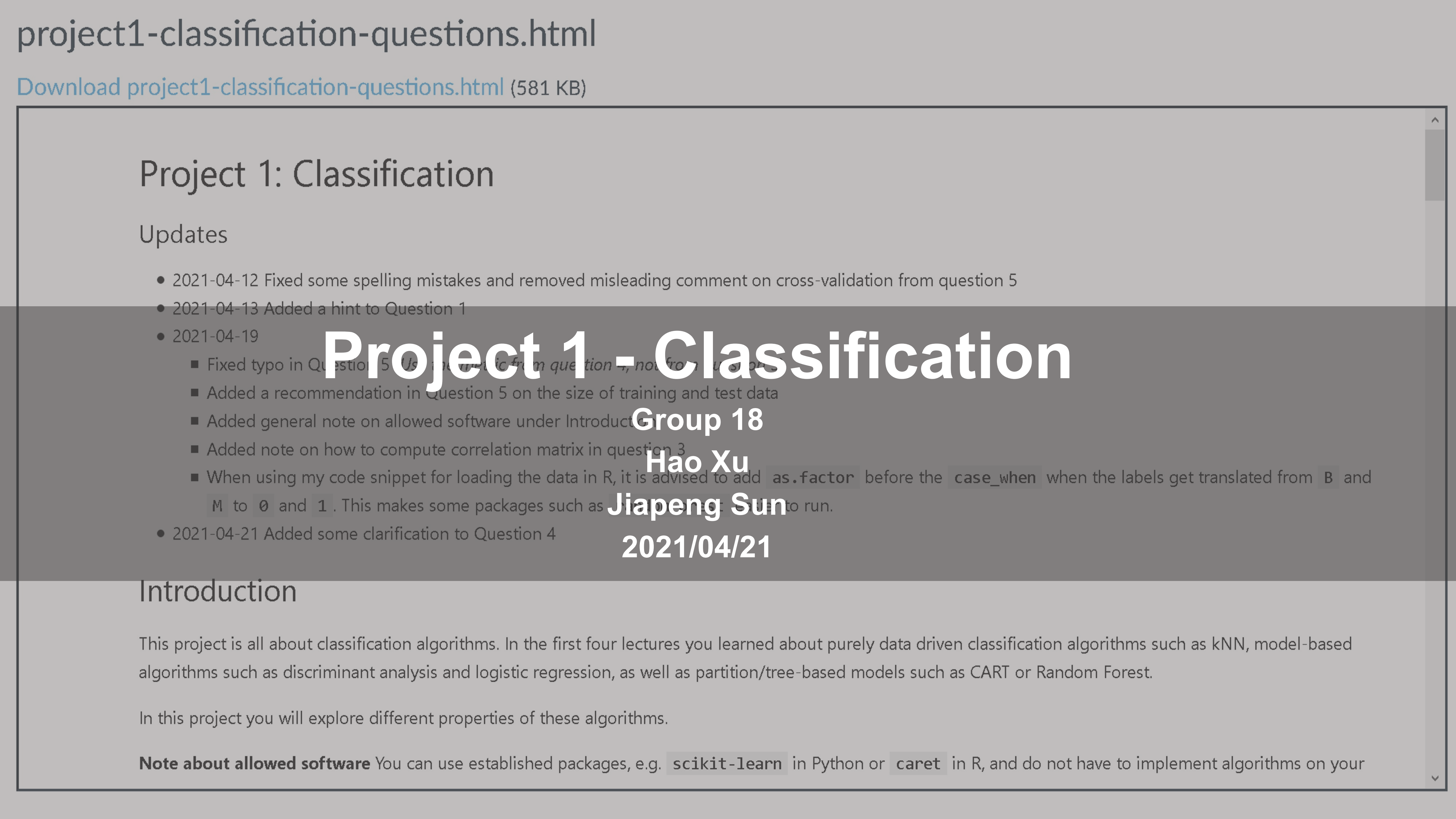
Project 1 - Classification

Group 18

Hao Xu

Jiapeng Sun

2021/04/21



project1-classification-questions.html

Download project1-classification-questions.html (581 KB)

Project 1: Classification

Updates

- 2021-04-12 Fixed some spelling mistakes and removed misleading comment on cross-validation from question 5
- 2021-04-13 Added a hint to Question 1
- 2021-04-19
  - Fixed typo in Question 5: *Use the metric from question 4, not from question 3*
  - Added a recommendation in Question 5 on the size of training and test data
  - Added general note on allowed software under Introduction
  - Added note on how to compute correlation matrix in question 3
  - When using my code snippet for loading the data in R, it is advised to add `as.factor` before the `case_when` when the labels get translated from `B` and `M` to `0` and `1`. This makes some packages such as [illegible] to run.
- 2021-04-21 Added some clarification to Question 4

Introduction

This project is all about classification algorithms. In the first four lectures you learned about purely data driven classification algorithms such as kNN, model-based algorithms such as discriminant analysis and logistic regression, as well as partition/tree-based models such as CART or Random Forest.

In this project you will explore different properties of these algorithms.

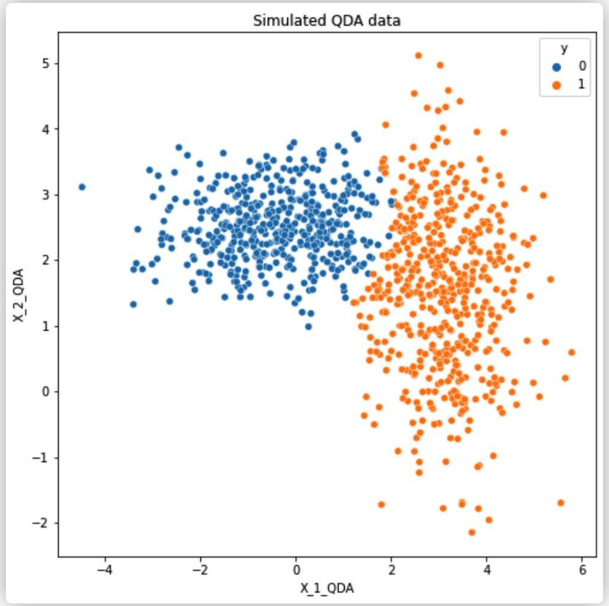
Note about allowed software You can use established packages, e.g. `scikit-learn` in Python or `caret` in R, and do not have to implement algorithms on your

Question 2

How we simulate data

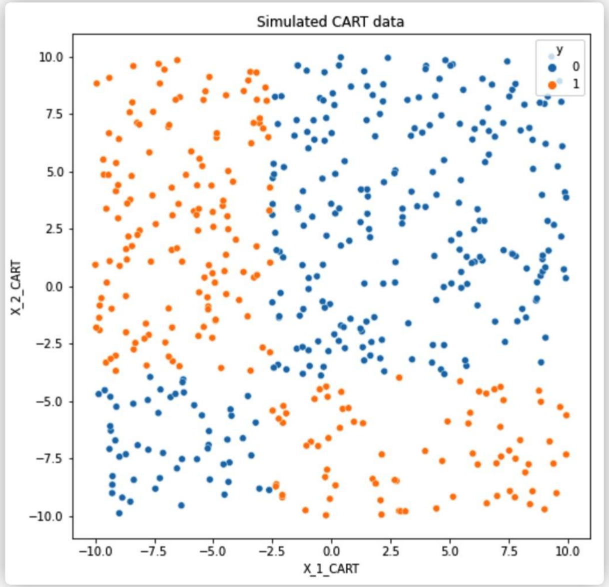
For QDA model:

1. We simulated 2 class situation.
2. For each class, fix a central point as mean, then fix the covariance matrix.
3. For each class, sample from Multivariate normal distribution with the fixed mean and covariance matrix.
4. Labeled the sample from above with the fixed class, and use such data to train the QDA classifier.
5. For each class, sample again from Multivariate normal distribution with the fixed mean and covariance matrix.
6. Labeled the sample from above with the prediction of trained QDA classifier, use those labeled data as simulated data for future use.



For CART model:

1. We simulated 2 class situation.
2. Fix a horizontal and a vertical line as the decision boundaries to separate the space into 4 zones.
3. For each class, fix 2 zones as its distribution space.
4. Generate random values for each feature of each sample, label such sample based one the zone it exist in.
5. Use those labeled data as simulated data for future use.



Question 2

Test error expectation

1. We expect QDA on CART dataset would show a limited performance, because the diagonal zone distribution of specific class is hard to be captured by the Multivariate normal distribution model.
2. We expect CART(RandomForest) on QDA dataset can show a nice performance, because the different Multivariate normal distribution can still be possible to be separated by sequence decision boundaries.

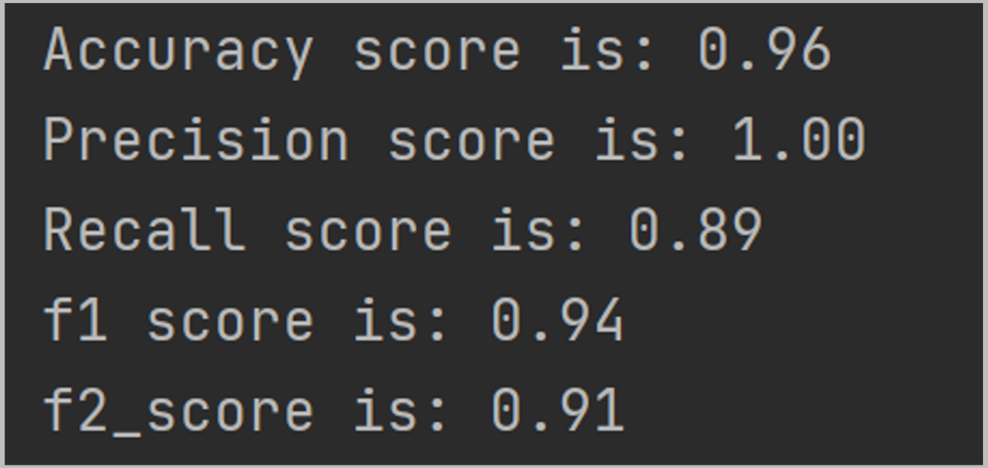
Test error comparing

1. We simulated 500 times for each dataset and prediction and take the average accuracy score as metric.
2. QDA on CART dataset shows a performance at about 92.8% accuracy.
3. CART on QDA dataset shows a performance at about 99.7% accuracy.
4. QDA shows a acceptable but not good enough performance on CART dataset, that might because the distribution of feature in different zones of CART dataset shows big difference with Multivariate normal distribution, especially when data are in diagonal zones, which can't be captured nicely by QDA model.
5. Random Forest shows a nice performance on QDA dataset, that might because the different Multivariate normal distribution data under QDA assumption still can be separated by the sequence decision boundaries, but we are not sure if this can still hold when the number of class grows.

Simulate data for the target method

- With the clear assumption of feature distribution under specific method, we can simulate the data from the distribution just as it assumed, which the target method will work best on.

Question 4 - 1



```
Accuracy score is: 0.96
Precision score is: 1.00
Recall score is: 0.89
f1 score is: 0.94
f2_score is: 0.91
```

Here, I choose 5 classification metrics:

- Accuracy
- Precision
- Recall
- F1
- F2

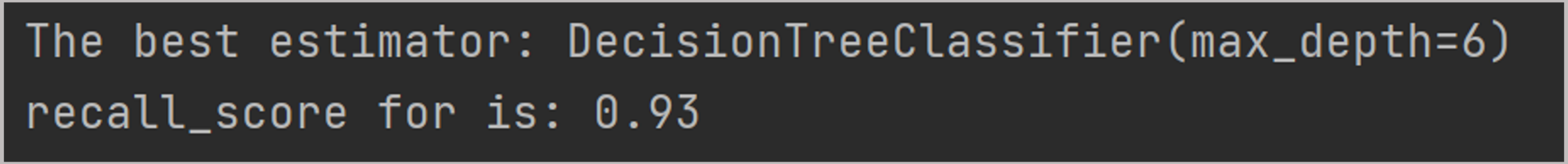
Precision score here is 1.00, which means when the model predicts it as positive, it is real positive; however, the recall score here is a little bit lower (0.89), which means there are 11% of real positive are predicted as negative.

Question 4 - 2

In this case, the goal of the machine learning project is to find out the malignant tumors, thus recall is more important than precision. Since if a patient without tumor is diagnosed as positive, that will just cost him some more money and time to take a further diagnosis. However, if a patient with tumor is diagnosed as negative, the consequence is much more severe. However, if the model always predicts positive labels, then the recall is 1, but it is obviously wrong.

Thus, we need to take both precision and recall into account. Here, we choose **recall** as our main metric, and then take f1 score into account.

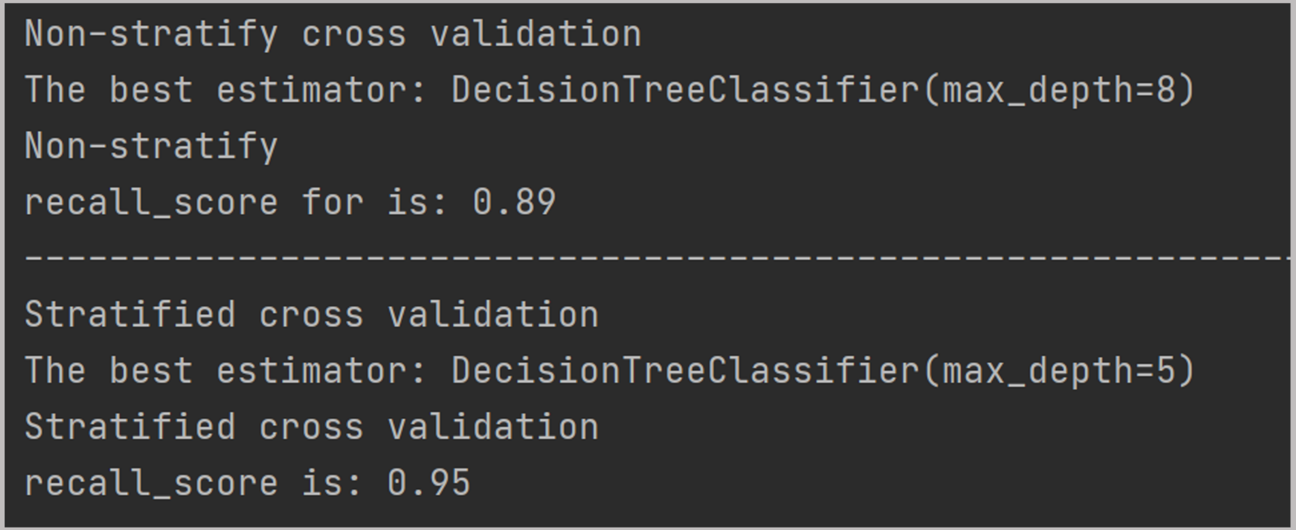
Question 4 - 3



```
The best estimator: DecisionTreeClassifier(max_depth=6)
recall_score for is: 0.93
```

Here, we choose Decision tree as an additional model. When the max_depth is 6, the recall score is 0.93, which is a little bit lower than Logistic regression.

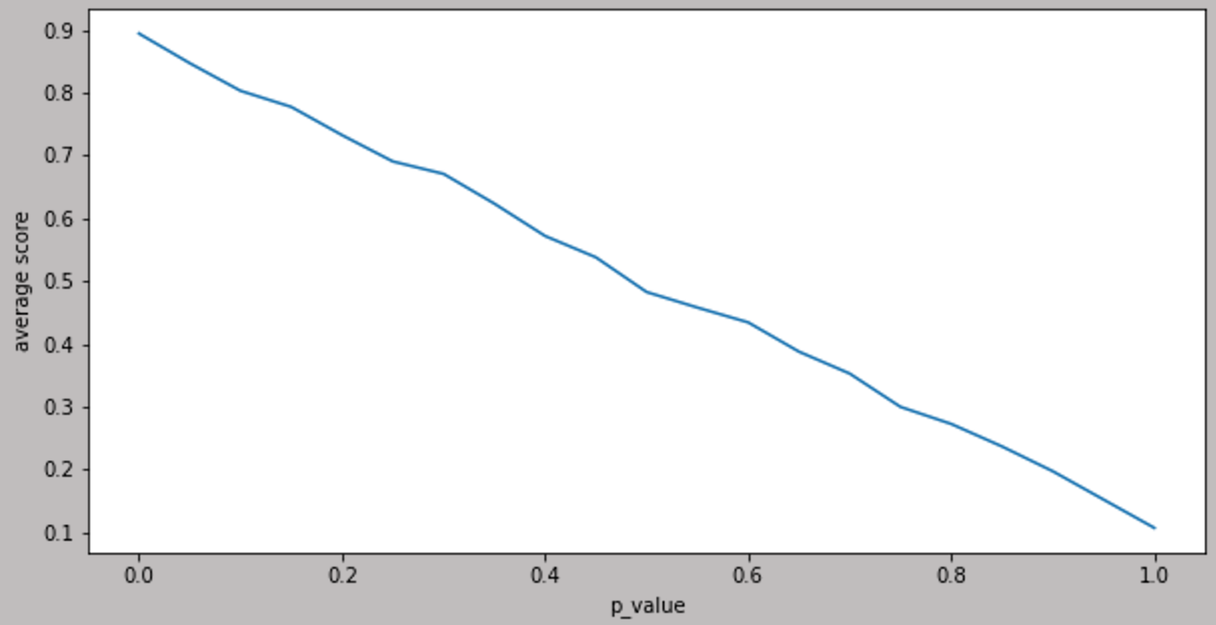
Question 4 - 4



```
Non-stratify cross validation
The best estimator: DecisionTreeClassifier(max_depth=8)
Non-stratify
recall_score for is: 0.89
--------------------------------------------------------------
Stratified cross validation
The best estimator: DecisionTreeClassifier(max_depth=5)
Stratified cross validation
recall_score is: 0.95
```

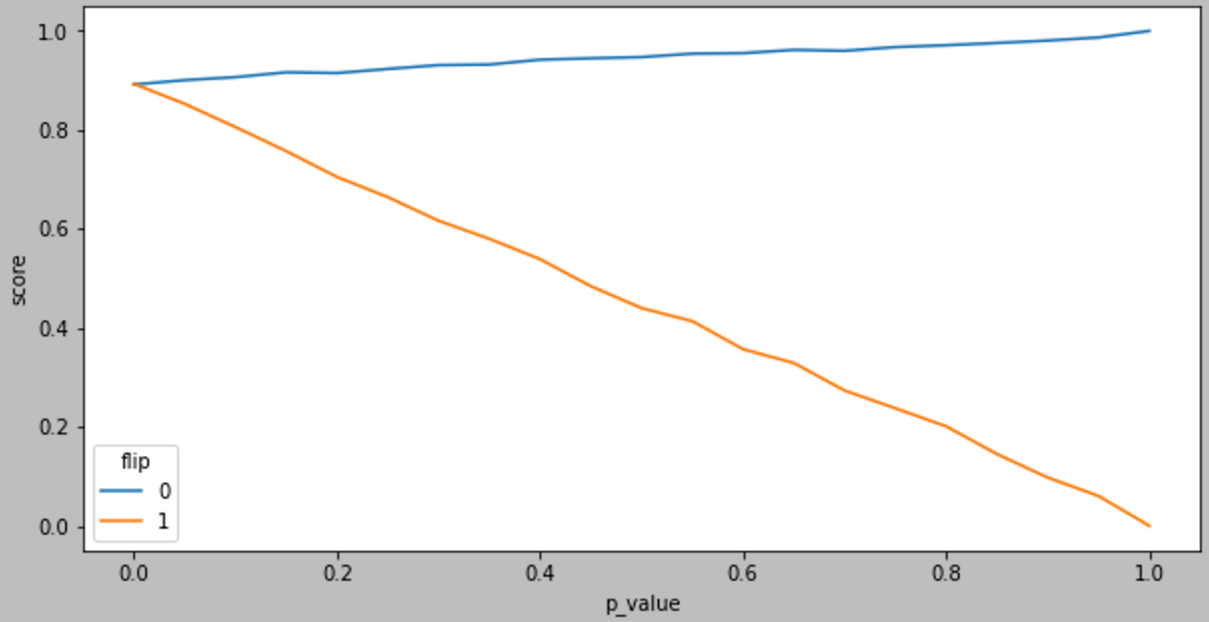
Because when using grid search for Decision tree model, every time the max_depth is not same, thus we tried many times. In most of them, the recall score of stratified cross validation is higher. That also makes sense in theory – for classification, we should use stratify to separate the folds, making the percentage of different labels in every fold same.

Question 5



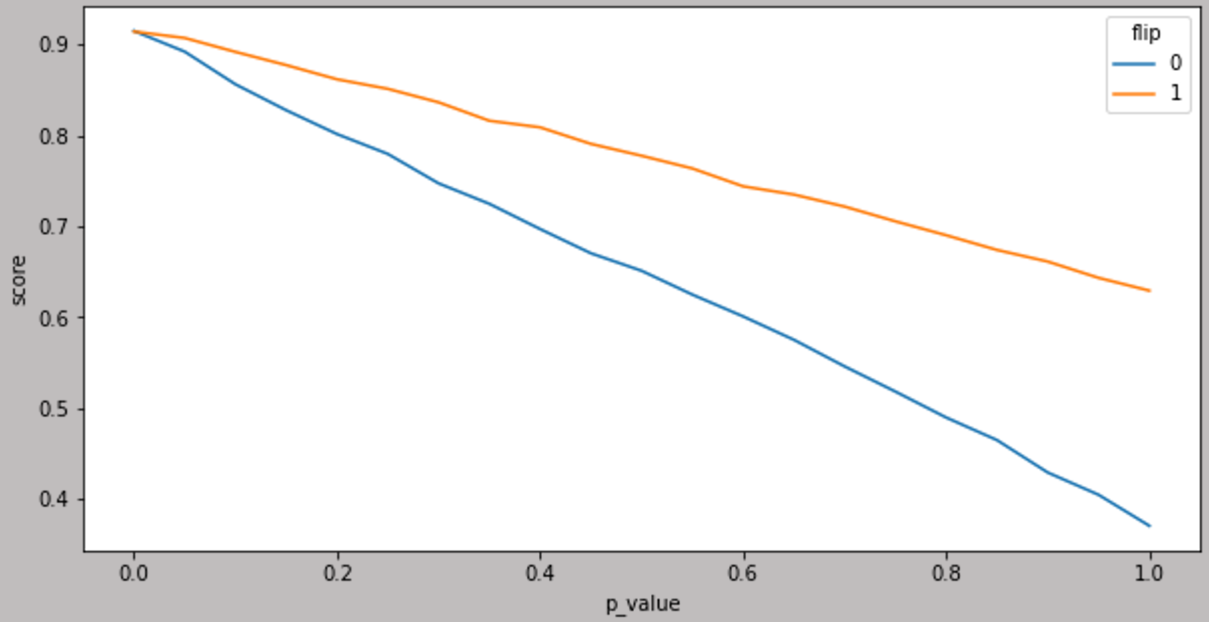
1. We set 20 p values between 0 to 1 and add 1 to the beginning as the unflipped score.
2. For each p value we repeat the split 5 times and for each split repeat the flip 20 times, so for each p value we have 100 result, and we take the average.
3. From the line plot we can see that the performance getting worse as the p value increase, p value and the score shows a linearity decreasing relationship.

Bonus question



1. As the p value increase, when flip 0 to 1, the classifier gives a better performance until the score close to 1 on prediction.
2. As the p value increase, when flip 1 to 0, the classifier gives a worse performance until the score close to 0 on prediction.
3. A possible reason is that out metric as recall focuses more on the positive class, by flipping 0 to 1 actually let the classifier more eager to predict all samples to positive, and that's why we got better and better result on such metric, the same reason applies why the performance getting worse when flip 1 to 0.

Bonus question



By checking the accuracy score, we found that both flip on 0 or 1 all leads to a worse performance, that gives the evidence of our assumption.

project1-classification-questions.html

Download project1-classification-questions.html (581 KB)

Project 1: Classification

Updates

- 2021-04-12 Fixed some spelling mistakes and removed misleading comment on cross-validation from question 5
- 2021-04-13 Added a hint to Question 1
- 2021-04-19
  - Fixed typo in Question 5 (*Use the metric from question 4, not from question 3*)
  - Added a recommendation in Question 5 on the size of training and test data
  - Added general note on allowed software under Introduction
  - Added note on how to compute correlation matrix in Question 3
  - When using my code snippet for loading the data in R, it is advised to add `as.factor` before the `case_when` when the labels get translated from `B` and `M` to `0` and `1`. This makes some packages such as `randomForest` easier to run.
- 2021-04-21 Added some clarification to Question 4

Thanks for Reading

Introduction

This project is all about classification algorithms. In the first four lectures you learned about purely data driven classification algorithms such as kNN, model-based algorithms such as discriminant analysis and logistic regression, as well as partition/tree-based models such as CART or Random Forest.

In this project you will explore different properties of these algorithms.

Note about allowed software You can use established packages, e.g. `scikit-learn` in Python or `caret` in R, and do not have to implement algorithms on your