Overall Scores: 48.5/50
Gradescope: 24 /24
Peer review: 4.5/5 (need to do it for each question)
Written answer: 20/21

Ali Lashgari
HW2 - STAT766
Fall 2022

Question 2 – part a

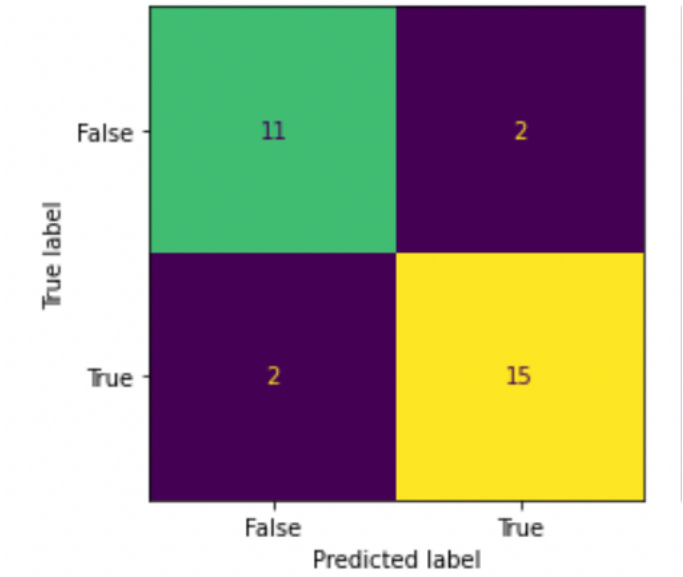
Using a classification threshold of 0.5 we have the following prediction for model 1 and model 2.

	target	Model1	predic_model1	Model2	predic_model2
0	false	0.1026	false	0.2089	false
1	false	0.2937	false	0.008	false
2	true	0.512	true	0.8378	true
3	true	0.8645	true	0.716	true
4	false	0.1987	false	0.1891	false
5	true	0.76	true	0.9398	true
6	true	0.7519	true	0.98	true
7	true	0.2994	false	0.8578	true
8	false	0.0552	false	0.156	false
9	false	0.9231	true	0.56	true
10	true	0.7563	true	0.9062	true
11	true	0.5664	true	0.7301	true
12	true	0.2872	false	0.8764	true
13	true	0.9326	true	0.9274	true
14	false	0.0651	false	0.2992	false
15	true	0.7165	true	0.4569	false
16	true	0.7677	true	0.8086	true
17	false	0.4468	false	0.1458	false
18	false	0.2176	false	0.5809	true
19	false	0.98	true	0.5783	true
20	true	0.6562	true	0.7843	true
21	true	0.9693	true	0.9521	true
22	false	0.0275	false	0.0377	false
23	true	0.7047	true	0.4708	false
24	false	0.3711	false	0.2846	false
25	false	0.444	false	0.11	false
26	true	0.544	true	0.3562	false
27	true	0.5713	true	0.92	true
28	false	0.3757	false	0.0895	false
29	true	0.8224	true	0.8614	true

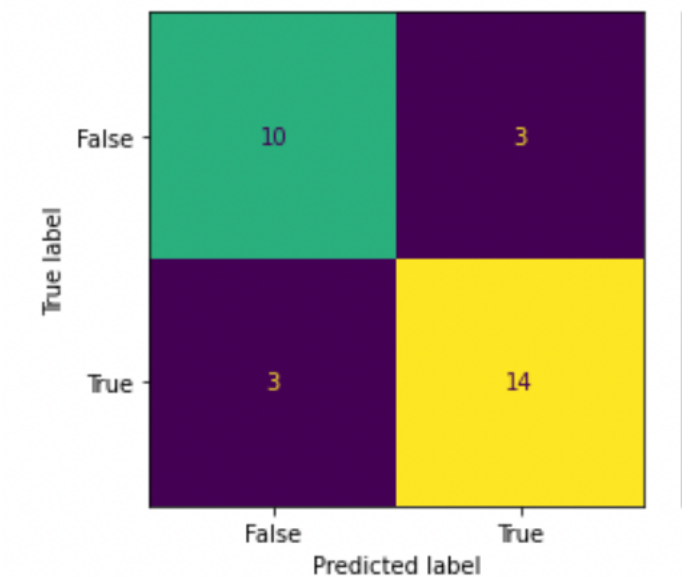
Considering the true is the positive target level. Here is the confusion matrix for each of the models.
Please note in this question:

classes= ('false', 'true')
class1_label= 'true'

Model 1:



Model 2:



Question 2 – part b

For model 1:

```
'TPR': 0.8823529411764706,
'TNR': 0.8461538461538461,
'FPR': 0.15384615384615385,
'FNR': 0.11764705882352941,
'precision': 0.8823529411764706,
'recall': 0.8823529411764706,
'F1': 0.8823529411764706,
'simple accuracy': 0.8666666666666667,
'average class accuracy-arithmetic': 0.8642533936651584,
'average class accuracy-HM': 0.8638743455497381
```

For model 2:

```
'TPR': 0.8235294117647058,
'TNR': 0.7692307692307693,
'FPR': 0.23076923076923078,
'FNR': 0.17647058823529413,
'precision': 0.8235294117647058,
'recall': 0.8235294117647058,
'F1': 0.8235294117647058,
'simple accuracy': 0.8,
'average class accuracy-arithmetic': 0.7963800904977376,
'average class accuracy-HM': 0.7954545454545455
```

Question 2 – part c

Model 1 is better because it has higher average class accuracy.

Question 2 – part d

For model one we have the following cumulative gain chart.

Note: The third column shows that we have class1(C1) which is 'true' or not. The prob should be sorted here.

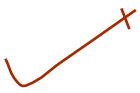
	target_y	prob	C1 or not	gain	cumulative_gain
19	false	0.9800	0	0.000000	0.000000
21	true	0.9693	1	0.058824	0.058824
13	true	0.9326	1	0.058824	0.117647
9	false	0.9231	0	0.000000	0.117647
3	true	0.8645	1	0.058824	0.176471
29	true	0.8224	1	0.058824	0.235294
16	true	0.7677	1	0.058824	0.294118
5	true	0.7600	1	0.058824	0.352941
10	true	0.7563	1	0.058824	0.411765
6	true	0.7519	1	0.058824	0.470588
15	true	0.7165	1	0.058824	0.529412
23	true	0.7047	1	0.058824	0.588235
20	true	0.6562	1	0.058824	0.647059
27	true	0.5713	1	0.058824	0.705882
11	true	0.5664	1	0.058824	0.764706
26	true	0.5440	1	0.058824	0.823529
2	true	0.5120	1	0.058824	0.882353
17	false	0.4468	0	0.000000	0.882353
25	false	0.4440	0	0.000000	0.882353
28	false	0.3757	0	0.000000	0.882353
24	false	0.3711	0	0.000000	0.882353
7	true	0.2994	1	0.058824	0.941176
1	false	0.2937	0	0.000000	0.941176
12	true	0.2872	1	0.058824	1.000000
18	false	0.2176	0	0.000000	1.000000
4	false	0.1987	0	0.000000	1.000000
0	false	0.1026	0	0.000000	1.000000
14	false	0.0651	0	0.000000	1.000000
8	false	0.0552	0	0.000000	1.000000
22	false	0.0275	0	0.000000	1.000000

Here is the cumulative gain chart for the second model.

	target_y	prob	C1 or not	gain	cumulative_gain
6	true	0.9800	1	0.058824	0.058824
21	true	0.9521	1	0.058824	0.117647
5	true	0.9398	1	0.058824	0.176471
13	true	0.9274	1	0.058824	0.235294
27	true	0.9200	1	0.058824	0.294118
10	true	0.9062	1	0.058824	0.352941
12	true	0.8764	1	0.058824	0.411765
29	true	0.8614	1	0.058824	0.470588
7	true	0.8578	1	0.058824	0.529412
2	true	0.8378	1	0.058824	0.588235
16	true	0.8086	1	0.058824	0.647059
20	true	0.7843	1	0.058824	0.705882
11	true	0.7301	1	0.058824	0.764706
3	true	0.7160	1	0.058824	0.823529
18	false	0.5809	0	0.000000	0.823529
19	false	0.5783	0	0.000000	0.823529
9	false	0.5600	0	0.000000	0.823529
23	true	0.4708	1	0.058824	0.882353
15	true	0.4569	1	0.058824	0.941176
26	true	0.3562	1	0.058824	1.000000
14	false	0.2992	0	0.000000	1.000000
24	false	0.2846	0	0.000000	1.000000
0	false	0.2089	0	0.000000	1.000000
4	false	0.1891	0	0.000000	1.000000
8	false	0.1560	0	0.000000	1.000000
17	false	0.1458	0	0.000000	1.000000
25	false	0.1100	0	0.000000	1.000000
28	false	0.0895	0	0.000000	1.000000
22	false	0.0377	0	0.000000	1.000000
1	false	0.0080	0	0.000000	1.000000

Question 2 – part e

The cut-off for model 1 is (0.8224, 0.23) and for model 2 is (0.9274, 0.23). Although the target p ercent (20%) is between the cumulative gain from 0.17 to 0.23, model 2 has the highest score cut -off (prob column) even before the cut-off row. So, I believe model 2 would perform better.



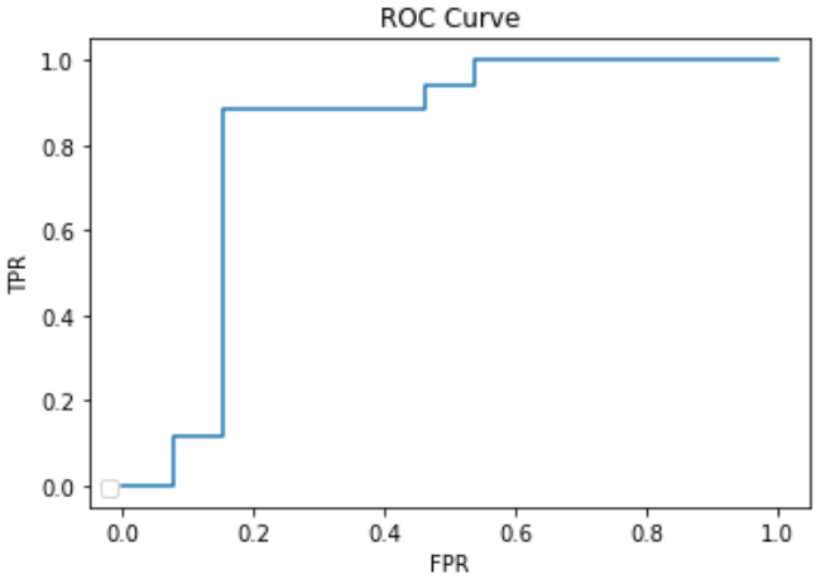
Should recommend Model 2 because you will need to pay more potential donors in order to capture the top 20% of real donors with model 1.

In an alternative way of thinking, with money giving to top 20% potential donors on the list, the model 2 will capture more true donors for extra donation.

Question 2 – part f

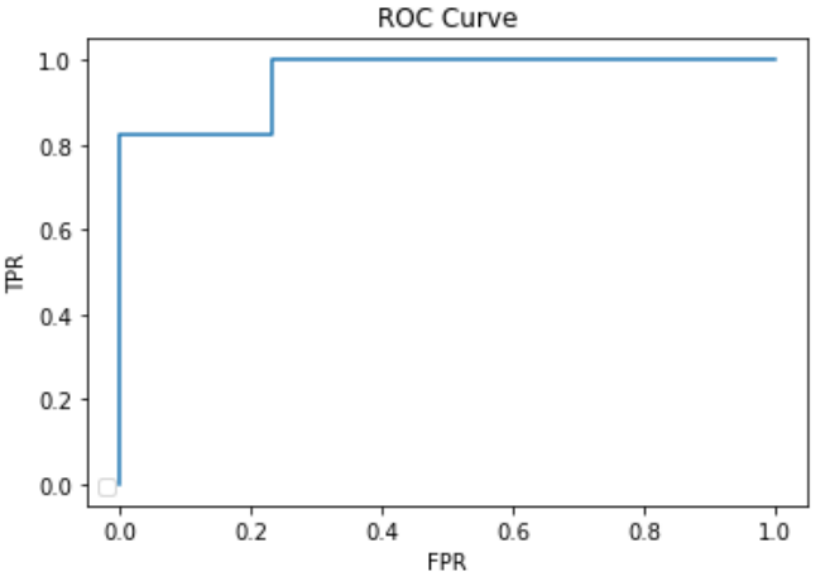
Here is the information for model 1:

```
'AUC': 0.8144796380090498,
'ROC_index': 0.8144796380090498,
'Gini_coef': 0.6289592760180995
```



And for model 2:

```
'AUC': 0.9592760180995474,
'ROC_index': 0.9592760180995474,
'Gini_coef': 0.9185520361990949}
```



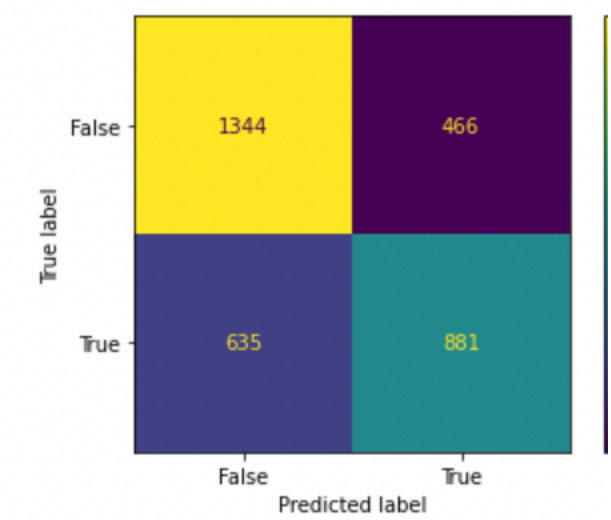
the area bellow ROC2 is greater, so model 2 performs better based on comparing AUC.

Question 3 – part a

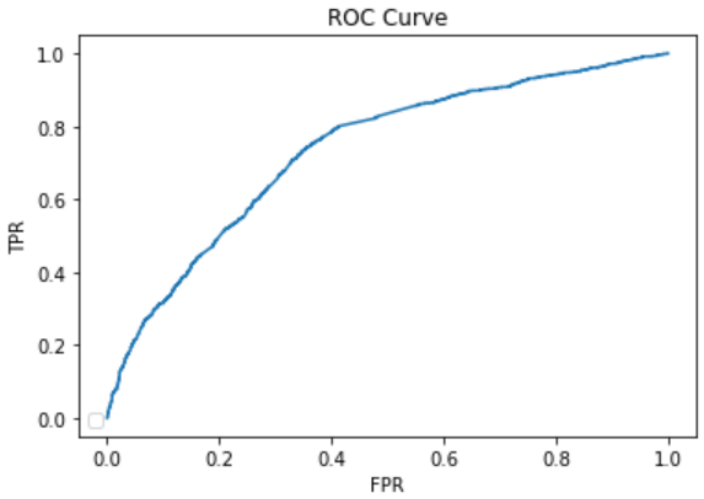
In this question, I consider:

classes = ("control","type2diabetes")
class1_label = "type2diabetes"

For TCF10foldCV_known_SNP_PC:



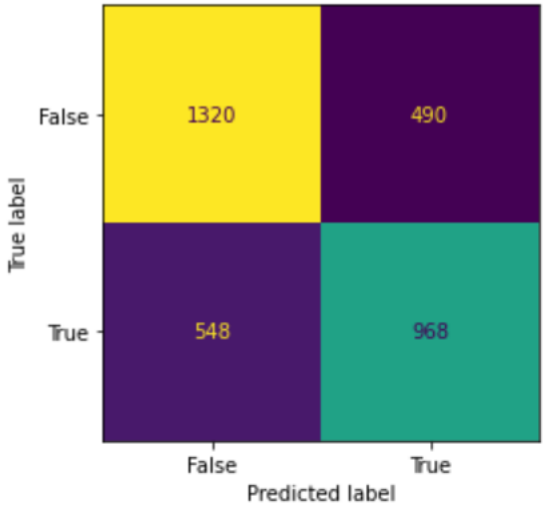

```
'TPR': 0.5811345646437994,
'TNR': 0.7425414364640884,
'FPR': 0.2574585635359116,
'FNR': 0.4188654353562005,
'precision': 0.6540460282108389,
'recall': 0.5811345646437994,
'F1': 0.6154383513796716}
simple_accuracy: 0.6689717378232111
'ave': 0.6618380005539439, 'HM': 0.6519971564769167

'AUC': 0.7340967069490809,
'ROC_index': 0.7340967069490809,
'Gini_coef': 0.4681934138981618
```



	target_y	prob	C1 or not	gain	cumulative_gain
1934	type2diabetes	0.964814	1	0.00066	0.000660
3251	type2diabetes	0.964211	1	0.00066	0.001319
639	type2diabetes	0.959630	1	0.00066	0.001979
1846	type2diabetes	0.959479	1	0.00066	0.002639
316	type2diabetes	0.959440	1	0.00066	0.003298
...	...	...	...	...	...
1841	control	0.110031	0	0.00000	0.999340
765	control	0.109635	0	0.00000	0.999340
1769	control	0.109588	0	0.00000	0.999340
3076	control	0.105584	0	0.00000	0.999340
3244	type2diabetes	0.103624	1	0.00066	1.000000

[3326 rows x 5 columns]

glmnet10foldCV_known_SNP_PC:

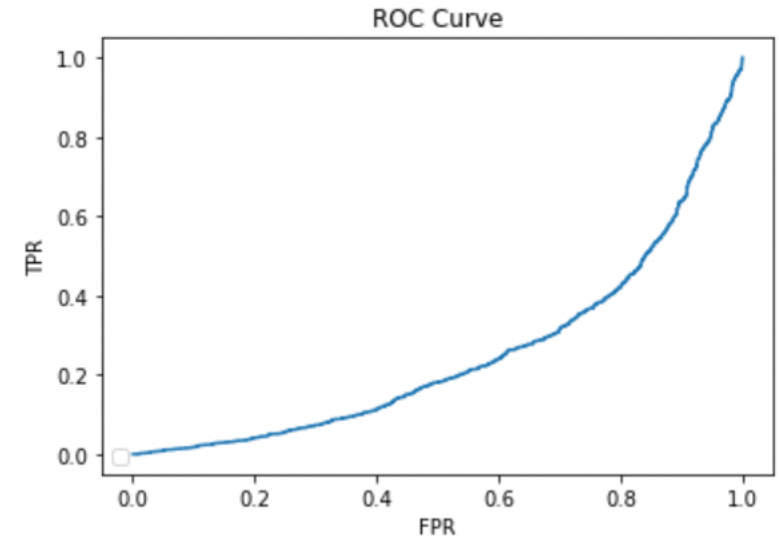


'TPR': 0.6385224274406333,
'TNR': 0.7292817679558011,
'FPR': 0.27071823204419887,
'FNR': 0.36147757255936674,
'precision': 0.663923182441701,
'recall': 0.6385224274406333,
'F1': 0.6509751176866174}

simple_accuracy: 0.687913409500902

'ave': 0.6839020976982172, 'HM': 0.6808909730363424

'AUC': 0.7479824778786863,
'ROC_index': 0.7479824778786863,
'Gini_coef': 0.49596495575737265



This curve is not correct.
You mistakenly flipped the probability of the two classes.

	target_y	prob	C1 or not	gain	cumulative_gain
1239	type2diabetes	0.929787	0	0.000000	0.000000
2518	type2diabetes	0.928307	0	0.000000	0.000000
316	type2diabetes	0.923263	0	0.000000	0.000000
2184	type2diabetes	0.922698	0	0.000000	0.000000
2956	type2diabetes	0.920426	0	0.000000	0.000000
...	...	...	...	...	...
152	control	0.123267	1	0.000552	0.997790
765	control	0.121492	1	0.000552	0.998343
2787	control	0.117939	1	0.000552	0.998895
1013	control	0.116792	1	0.000552	0.999448
2381	control	0.096212	1	0.000552	1.000000

[3326 rows x 5 columns]

Question 3 – part b

All the performance measures show that second model has the better performance. It not only has the higher simple and average accuracy but also has the higher AUC.

Question 4

If we have the same number of samples per class, accuracy can be a useful measure; however, if we have an imbalanced set of samples, accuracy is useless. Furthermore, a test with a high accuracy may perform worse than a test with a lower accuracy. That's why we can use the average class accuracy (for instance like a k-NN and a naive Bayes example in the lecture note).

Insufficient.

The average accuracy for classification, precision, recall and F1 score might not be sufficient to evaluate the performance of a model as they already assume a fixed threshold.

For this reason, we need the AUC, ROC index, Gini coefficient and KS statistic to evaluate the performance under different thresholds for classification. When all the above performance measures are close for two or more models, KS statistic could help determine the model that most effectively separates the classes in any classification problem. Also, in cases when the resources are limited for model deployment, cumulative gains chart can give an appropriate model choice to limit the resources used for model deployment.

In summary, a single accuracy measure cannot alone determine the performance of a model, and in cases when choosing the best model out of many, different metrics are required to compare performances.