Delft University of Technology

Data Science for Finance
WI3435TU

Assignment: Part III. Trees

Authors:
Dinu Gafton
Dan Sochirca

January 29, 2023

Question 1. A single decision tree model

a) The information gain for the root node of a tree based on the five most important features

Using logistic regression in the previous assignment helped us identify the **5 most important features**, which are: **sub_grade, term, dti, fico_range_low, loan_amnt**.

The figure below presents the information gain for the root node. It is clear from the results that the feature **sub_grade** has the highest information gain out of the rest of the features. This implies that the **split in the root node should be done using this feature**, and the threshold for the split should be **around -0.08**. As it is reported for question 1b below, this is indeed what the classifier did.

	Feature	Threshold value	Expected entropy	Information gain
0	sub_grade	-0.082786	0.735665	0.039817
1	term	-0.525092	0.757793	0.017689
2	dti	0.122575	0.768440	0.007043
3	fico_range_low	0.311620	0.762919	0.012564
4	loan_amnt	-0.535526	0.772372	0.003110

Figure 1: Expected entropy and information gain for the root node. The second column displays the threshold value that maximizes the information gain for each of the features.

b) The decision tree and it's performance for different probability thresholds

We have trained a decision tree using the 5 most important features outlined above, with a depth of 5 and using entropy as the information gain measure. A plot of this tree is provided in the **appendix A**, figure 11.
The plot reveals that **the split in the root node is done for feature sub_grade using a threshold of -0.005.** This corresponds to our expectations described in question 1a.

Next, we considered **different probability thresholds** in order to improve the classification, and applied them to the estimated probabilities. The thresholds we used are 0.5, 0.6, 0.7, 0.75, 0.80, and 0.85. **The ROC curve** and **confusion matrices** for different thresholds are presented in **appendix A**, figure 12.

METRICS:

THRESHOLD	0.5	0.6	0.7	0.75	0.8	0.85
accuracy	0.773215	0.764004	0.735065	0.683905	0.5993	0.50151
true pos rate	0.986513	0.919244	0.837116	0.724292	0.562159	0.399242
precision	0.778428	0.803024	0.822294	0.84341	0.872529	0.896466
AUC-score	0.521562	0.580851	0.614663	0.636255	0.64312	0.622166

Figure 2: Different metrics corresponding to the different thresholds considered. The highest AUC score is highlighted in yellow.

In the table depicted above, we computed different metrics for each of the threshold values. The best choice of threshold in terms of the AUC is **0.8**, with **AUC= 0.64312**. The model using this probability threshold also has: **TPR=0.562159** and **precision=0.872529**. As mentioned before, the ROC curve and confusion matrix for this threshold can be found in **appendix A**, figure 12.

c) Analysis of a decision tree using the ten most important features

Our next decision tree was trained on the **10 most important features**, which were also identified using regression:
sub_grade, term, loan_amnt, dti, fico_range_low, mort_acc, mo_sin_old_rev_tl_op, earliest_cr_line, acc_open_past_24mths, int_rat.

	Feature	Threshold value	Expected entropy	Information gain
0	sub_grade	-0.082786	0.735665	0.039817
1	term	-0.525092	0.757793	0.017689
2	loan_amnt	-0.535526	0.772372	0.003110
3	dti	0.122575	0.768440	0.007043
4	fico_range_low	0.311620	0.762919	0.012564
5	mort_acc	-0.315690	0.769228	0.006254
6	mo_sin_old_rev_tl_op	-0.482994	0.772289	0.003193
7	earliest_cr_line	0.324473	0.773879	0.001603
8	acc_open_past_24mths	-0.324224	0.769280	0.006203
9	int_rate	-0.064520	0.735471	0.040011

Figure 3: Expected entropy and information gain for the root node (of the decision tree trained on 10 features). The highest information gain is highlighted in yellow.

In figure 3, we show the information gain for the root node. The feature **int_rate** has the highest value (highlighted in yellow), for a threshold of **-0.06**. Indeed, the decision tree we trained on the 10 most important features has chosen this same feature for the split in the root node, with the same threshold of -0.06. A plot of the tree can be found in **appendix A**, figure 13.

METRICS:

THRESHOLD	0.5	0.6	0.7	0.75	0.8	0.85
accuracy	0.774246	0.765464	0.723464	0.661156	0.597967	0.475414
true pos rate	0.981321	0.92166	0.80335	0.674983	0.558077	0.357313
precision	0.78156	0.803073	0.832014	0.854768	0.87493	0.903775
AUC-score	0.529937	0.581181	0.629212	0.644842	0.645029	0.614751

Figure 4: Different metrics corresponding to the different thresholds considered for the decision tree trained on 10 features. The highest AUC score is highlighted in yellow.

As done before, different metrics were computed for different probability thresholds, and the results are diplayed in figure 4. Again, **the best threshold** in terms of AUC is **0.8**, with **AUC=0.645029**. The **precision** is **0.87493** and **TPR= 0.558077**

The ten-feature tree model and the five-feature each perform very similar to each other, but the ten-feature tree is **slightly better** in terms of AUC and precision. (The difference in AUC is 0.002 and the same is for the precision). Although the difference in performance is negligible, **the ten-feature tree model is the best out of the two**, if we're being objective. The confusion matrix and ROC is provided in **appendix A**, figure 14.

Question 2. Tree ensemble methods

A Random Forest is a collection of Decision Trees, where each tree is trained on a random subset of the data. Using a Random Forest can improve the overall accuracy of the model and also reduce the overfitting problem, which is common in Decision Trees.

For the majority of the parameters of the *RandomForestClassifier* class, we have decided to leave the default values, but we have also changed some of them as it can be seen below.

```
random_forest = RandomForestClassifier(criterion='entropy',oob_score=True,max_samples=0.3,max_depth=5)
```

Figure 5: Changed parameters for the RandomForestClassifier object

We have changed the criterion to **entropy**, as it was required in the assignment, the **oob_score** was set to True (it is the parameter that will allow us to generate and see the *out-of-bag errors*), **max_samples** was set to 0.3 (meaning 30% of the training set will be sampled to evaluate and generate the OOB scores for each Decision Tree).

Optimization

For this task, we have chosen different values for the **max_depth** (meaning the maximum depth of a tree) and **max_features** (meaning the number of features to consider when looking for the best split) to find the optimal combination of these parameters that we will further work with.

Figure 6 shows us the OOB errors of the different combinations. It can be seen that the model with **max_depth = 12** gives the lowest OOB errors (for both combinations). Thus, we have decided to use the combination of **max_depth = 12** and **max_features = 0.3** as the optimal parameters for our model.

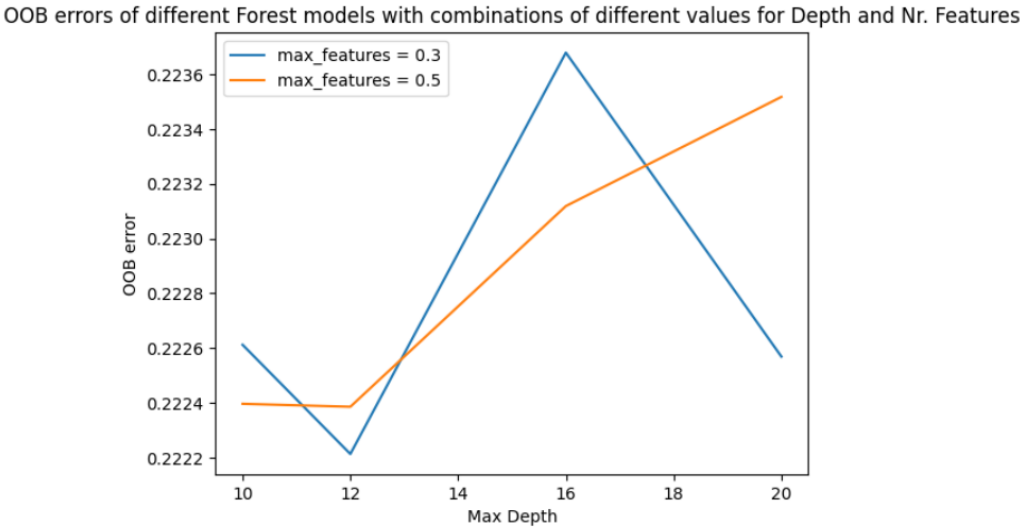

Figure 6: OOB errors for models with different combinations of Max Depth and Max Features

Using the newly optimised parameters, we decided to study the effect of number of trees in the Random Forest (**n_estimators**). By choosing between different values ([100, 110, 120, 130]) we have evaluated the model as before and gained the following results:

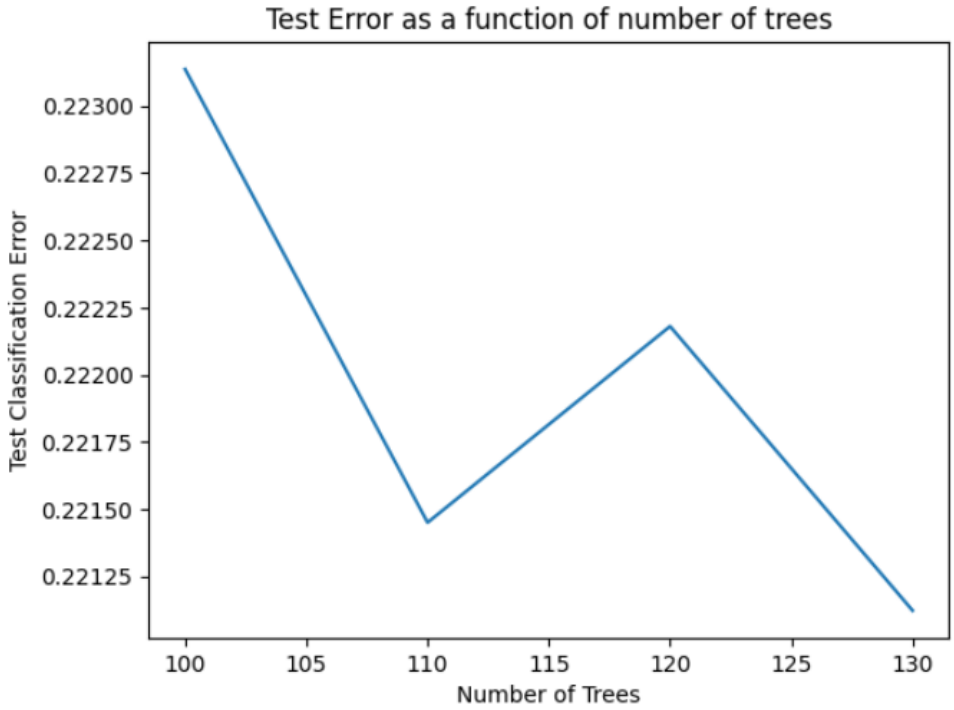

Figure 7: Test Error as a function of number of trees

The difference in performance is insignificant (all the errors are between 0.221 and 0.223). Therefore, we decided to use **n_estimators = 120** as the optimal parameter.

Best Random Forest model

Using the optimised parameters (**max_depth = 12**, **max_features = 0.3** and **n_estimators = 120**, we evaluated the model again and received the following results:

```
OOB error for best model =  0.22250382864908003 ; Test error for best model =  0.2216518194171826
```

Figure 8: OOB error and Test error for the best model

We can see that the OOB error and Test error are **extremely close**, meaning that the model performs as well on the training samples as on the test set (the unseen data) and we do not have an overfitting problem.

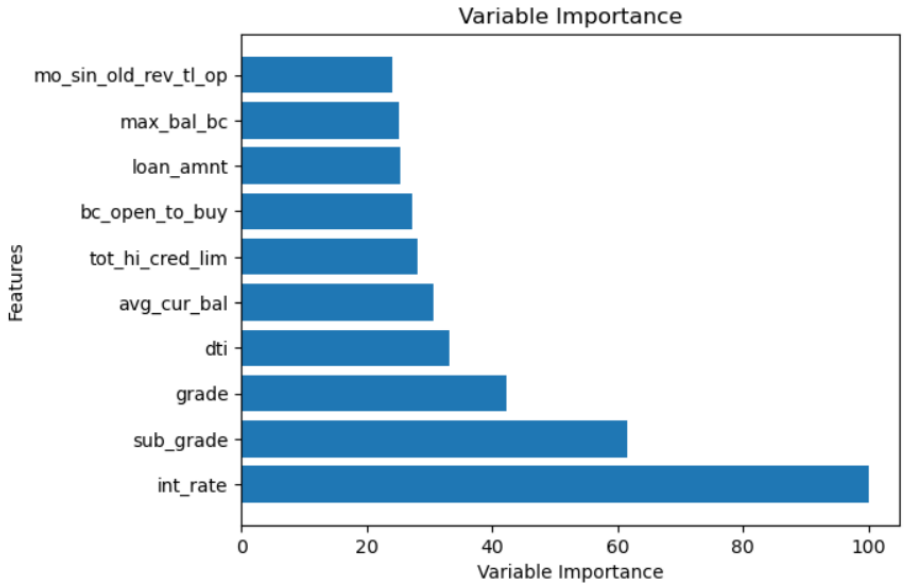

Figure 9: Variable Importance for the best Random Forest model

Figure 9 shows the Variable Importance chart for our best Random Forest model. Since there are 53 features available, only the 10 most important features were included in the chart. It can be seen that some of them are also considered as important for the Regression models.

Question 3. The best model

As usual, to find the best model we need to find the probability threshold that maximizes AUC. We did so, for our random forest classifier, and it appears to be **0.75**, with **AUC=0.652782**.

Again, we computed different metrics and displayed them in figure 10 below. The ROC curve can be found in **appendix C**, figure 15. The model has a **higher AUC** (by roughly 0.01) than the one we found in question 1c, which had an AUC of 0.645029.

```
Confusion matrix of the best random forest model:
          [[0.14487392 0.0845035 ]
          [0.24538225 0.52524032]]
```

	accuracy	true pos rate	precision	AUC-score
0	0.670114	0.681579	0.861411	0.656588

Figure 10: Different metrics computed for the random forest classifier with a probability threshold of 0.75.

Therefore, we conclude that the best model is **a random forest classifier with a probability threshold of 0.75.**

Appendix A: Additional images for question 1

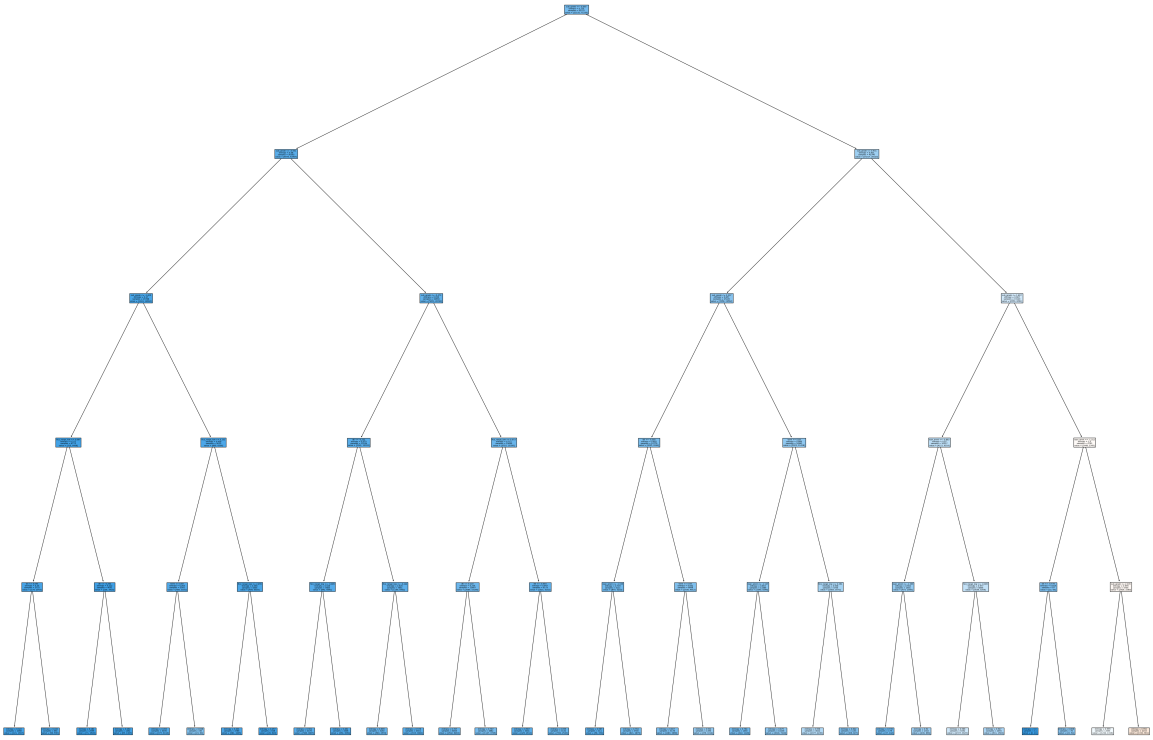

Figure 11: The decision tree for **question 1b**. The tree is based on the **5** most important features, has a depth of 5 and the split in the root node is done for feature *sub_grade* using a value threshold of -0.005, with entropy= 0.775.

```
Confusion matrix for threshold = 0.5
 [[0.01298505 0.21639237]
 [0.01039307 0.7602295 ]]

Confusion matrix for threshold = 0.6
 [[0.05561427 0.17376315]
 [0.06223262 0.70838995]]

Confusion matrix for threshold = 0.7
 [[0.08996427 0.13941316]
 [0.12552217 0.64510041]]

Confusion matrix for threshold = 0.75
 [[0.12574865 0.10362877]
 [0.21246666 0.55815592]]

Confusion matrix for threshold = 0.8
 [[0.16608788 0.06328955]
 [0.33741004 0.43321254]]

Confusion matrix for threshold = 0.85
 [[0.19384468 0.03553274]
 [0.46295737 0.30766521]]
```

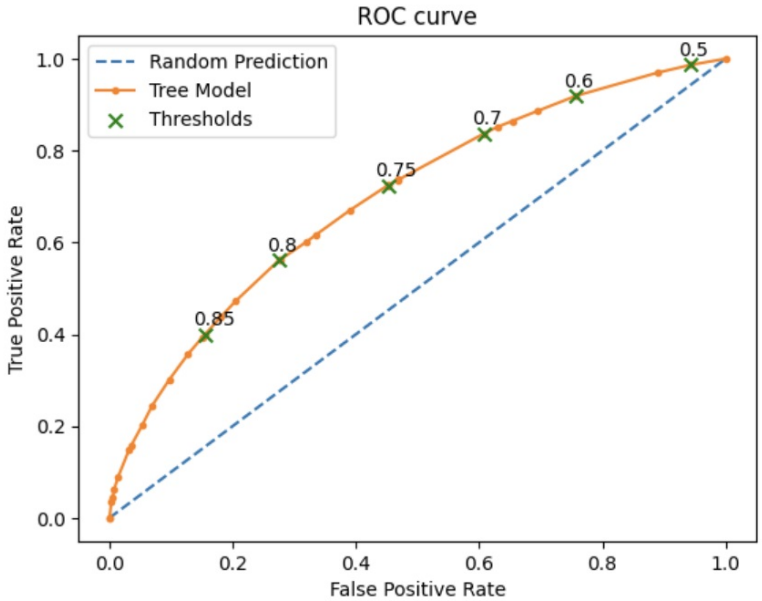

Figure 12: The confusion matrices for different thresholds and ROC curve of the tree for **question 1b**. The decision tree is based on the 5 most important features.

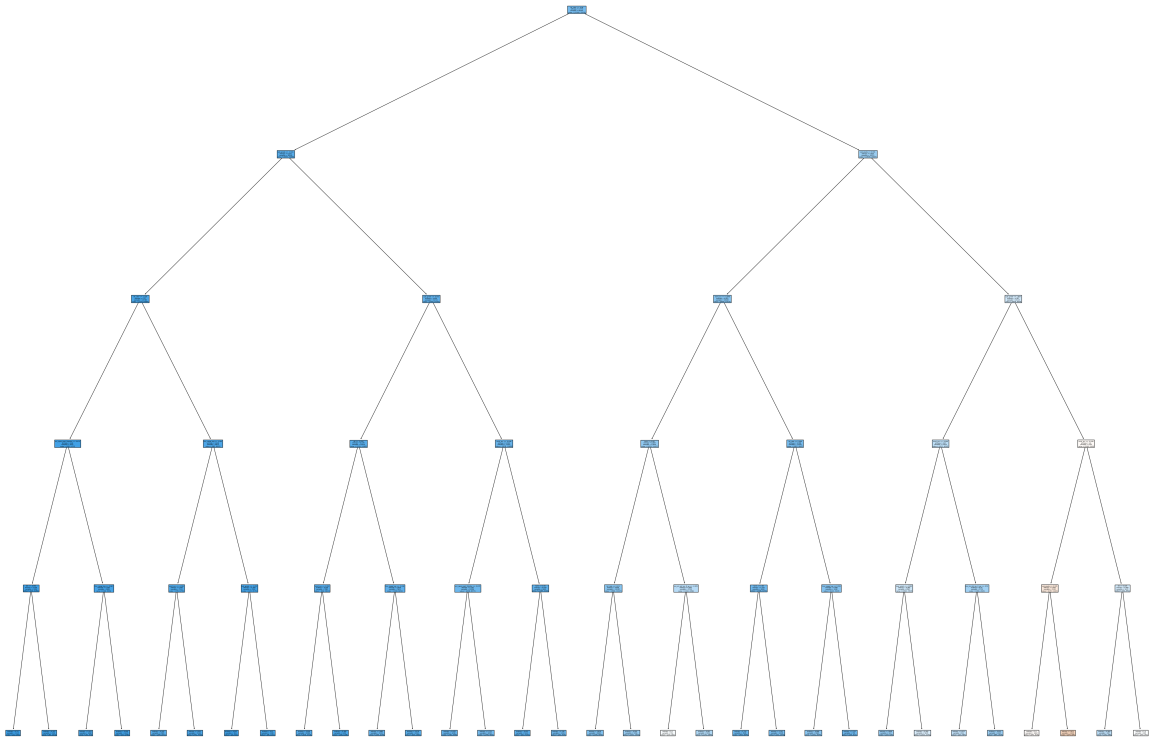

Figure 13: The decision tree for **question 1c**. The tree is based on the **10** most important features, has a depth of 5 and the split in the root node is done for feature *int_rate* using a value threshold of -0.06, with entropy= 0.775.

```
Confusion matrix for threshold = 0.8
[[0.16789974 0.06147768]
[0.34055564 0.43006694]]
```

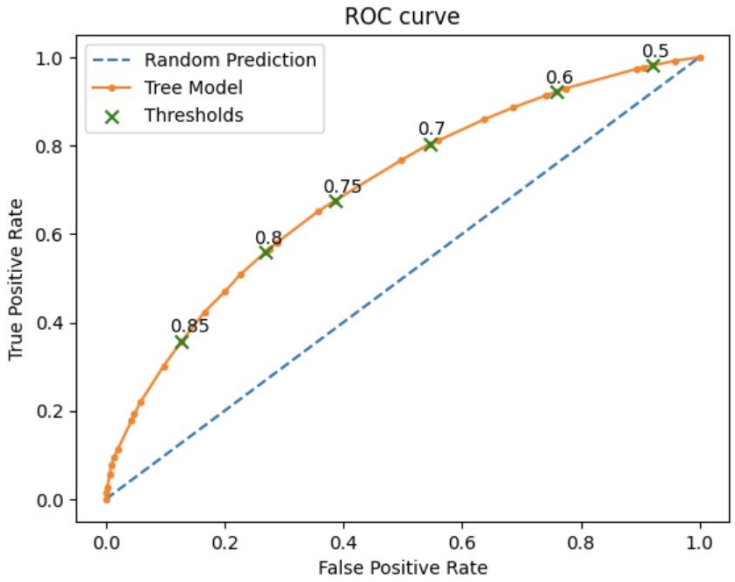

Figure 14: The confusion matrix and ROC curve of the best tree model in **question 1c**. The decision tree is based on the 10 most important features.

Appendix C: Additional images for question 3

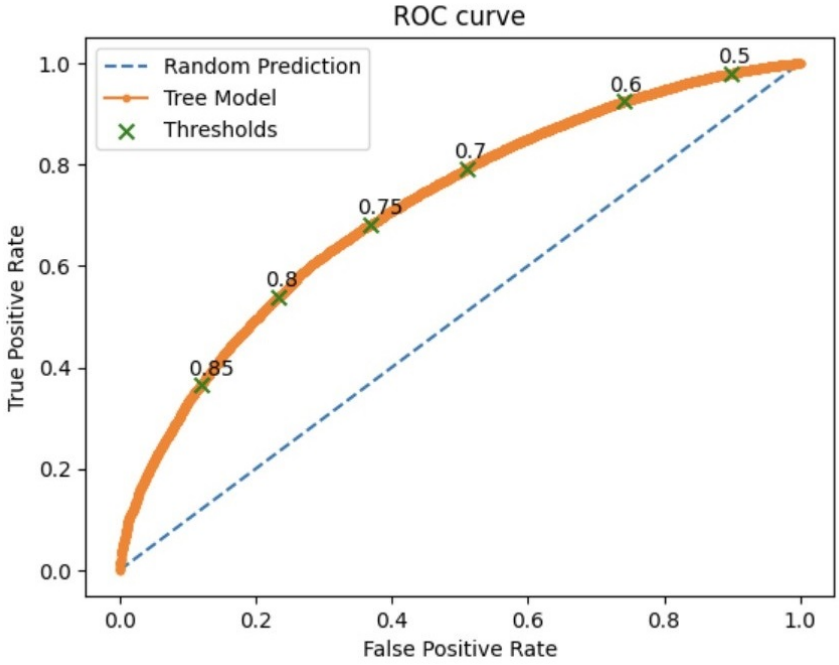

Figure 15: The ROC curve of the best model, which is a random forest classifier with a probability threshold of 0.75.