# **Project 2**

**Name: Caleb Thian Jia Le 田家樂**

**Student ID: NN6114035**

## About Data

The data created is about online courses. 10000 online courses are generated with 11 features, which are subject, subscribers, free, fee, reviews, avg. reviews, level, lectures, duration, published date and subtitles. The features’ definition and the way to generate them is listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Features | Definition | Way to generate | Graph |
| subject | The course’s involved area, expected only 1 value | Random choose among {Management, Photography, Social Sciences, Science, Personal Development, Design, Information Technology, Music} |  |
| subscribers | The number of subscribers of the course. | Sampling from a normal distribution N(7500,2500), each sample must no less than 0 |  |
| free | If the course is a free course, set as 1. | Bernoulli Trial with p=0.25 |  |
| fee | The fee of the course. | 0 if free, the others sample from exponential distribution (Exp(0.01)+10)\*10 |  |
| reviews | The number of reviews. | Sampling from uniform distribution with max = subscribers |  |
| avg. reviews | The average score from reviews. | Sampling from uniform distribution 0.5n, 0⩽ n ⩽10 and n is integer. (Note that the plot merges 4.5 and 5.0 into a bar, it is still a uniform distribution) |  |
| level | The level of the course. | Random choose among {Beginner, Intermediate, Expert} |  |
| lectures | The number of lectures provided. | Sampling integer > 0 from normal distribution N(10,5). |  |
| duration | The total duration of the lectures in minutes | Sampling average duration > 0 from normal distribution N(30,10) and multiply with number of lectures |  |
| published date | The published date of the course | Generate a random date between 2002.01.01 and 2021.12.31 |  |
| subtitles | If the course has subtitles, set as 1. | Bernoulli Trial with p=0.5 |  |

This data will like to recommend several courses according to these features. If a course will be recommended, at least 1 of the following rule should be satisfied:

1. subscriber > 12000
2. review >= 0.8\*subscriber and avg. reviews >= 4.5
3. level = beginner and fee <= 100
4. level = intermediate and fee <= 250
5. level = expert and fee <= 500

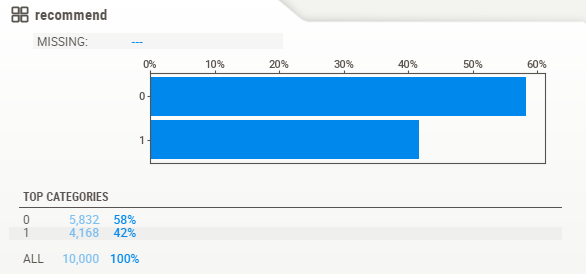


Figure 1: Distribution of the course

By these rules, the ratio of course that be recommended and not recommended is about 3:2. The plot below shows its distribution, 1 indicates the course is recommended.

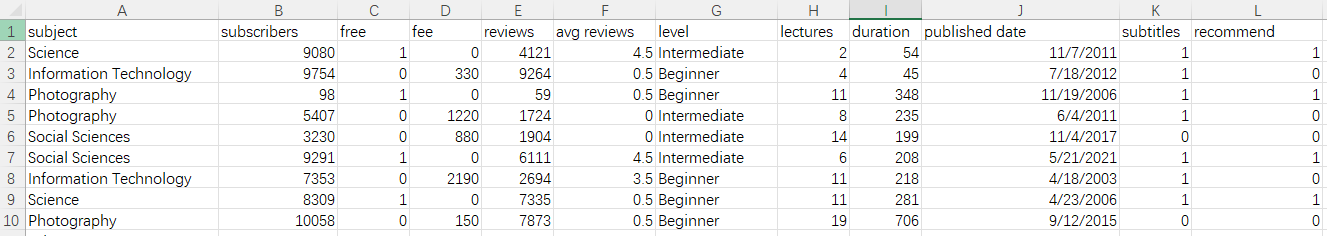


Figure 2: Example data generated

The details of the data can be viewed in EDA.html. The data is split into 4:1, i.e., 8000 for training and 2000 for testing.

## Data Pre-processing

Several data pre-processes are done on the dataset.

### Label Encoding

Label encoding on the categorical features, which are subject and level.

### Conversion

Convert the published date into timestamp so that classification can be run on it. Reviews also be converted into ratio but not integer. The reason to do this conversion will be discussed in the session discussing decision tree.

### Normalization

Normalization is done on numerical features, even though it doesn’t affect decision tree, it will improve the result of KNN and Naïve Bayes classifier.

## Result

Decision tree, KNN, and Naïve Bayes classifiers are tried to classify these courses. Discussion for each classifier will be made at each session and comparison will be made at the end of each session.

### Decision Tree

The below is the tree generated by decision tree classifiers with the result. But because the tree is too large, for details please refer to *decision\_tree\_no\_conversion.svg.*

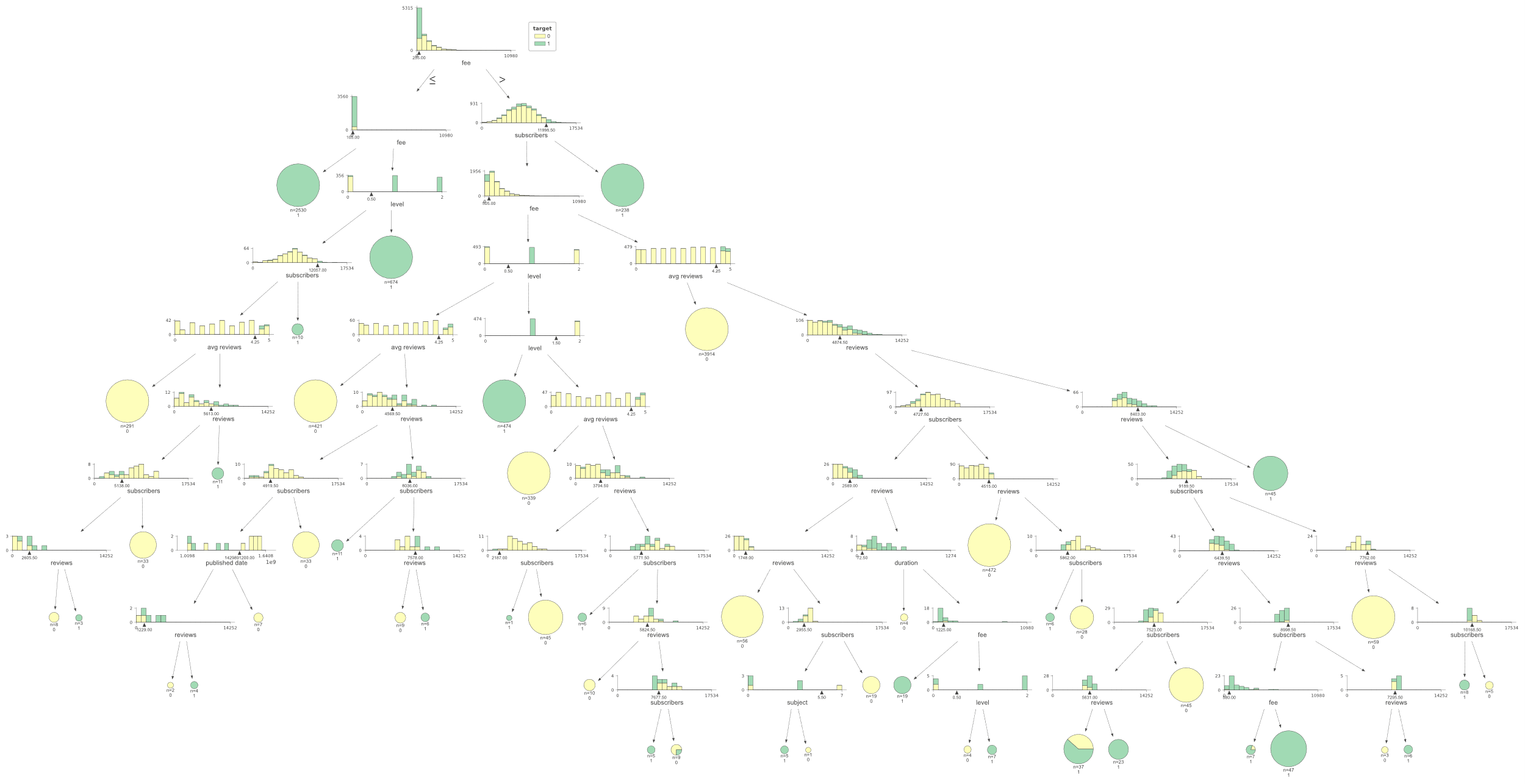


Figure 3: Decision Tree generated without conversion of reviews

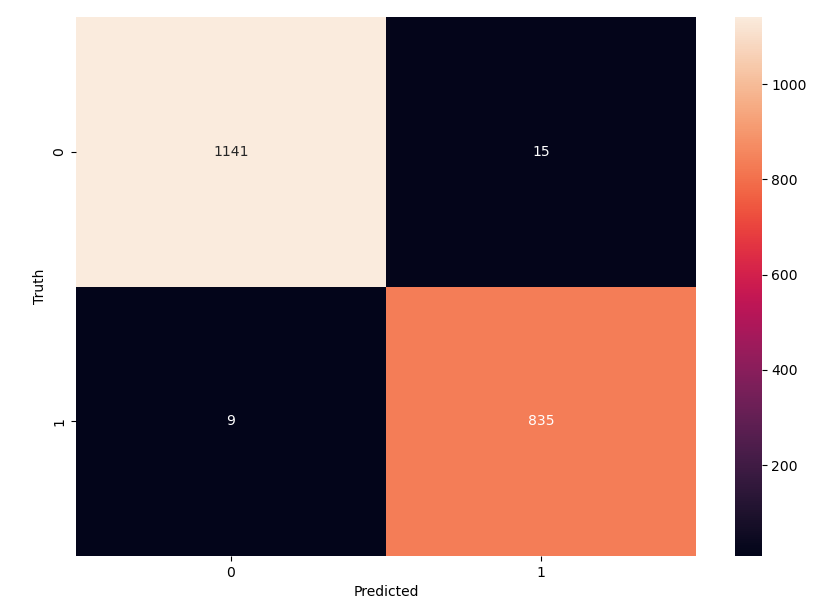


Figure 4: Confusion matrix of decision tree without conversion of reviews

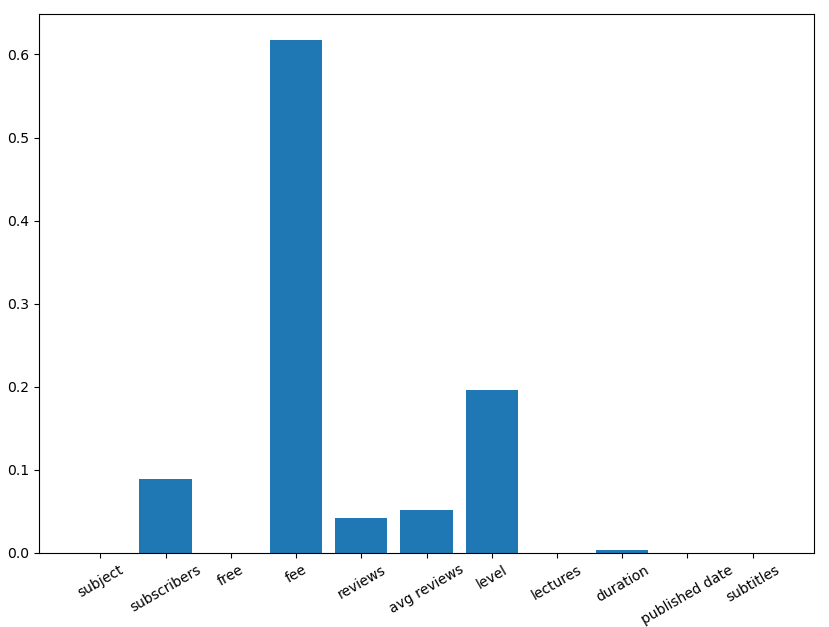


Figure 5: Feature importance of decision tree without conversion of reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.99 | 0.99 | 0.99 | 1156 |
| 1 | 0.98 | 0.99 | 0.99 | 844 |
| accuracy |  |  | 0.99 | 2000 |
| macro avg | 0.99 | 0.99 | 0.99 | 2000 |
| weighted avg | 0.99 | 0.99 | 0.99 | 2000 |

Table 1: Classification Report of Decision Tree without conversion of reviews

Notice that the rule generated by this decision tree classifiers is too complex, thus it is not listed in this report. For details, please refers to *decision\_tree\_no\_conversion.log*. By listing out the wrong cases, observed that 15 cases should not be recommended, and 9 cases are not recommended because the classifiers did not detect that the cases satisfied the rule 3 which is “review >= 0.8\*subscriber and avg. reviews >= 4.5”. Thus, conversion is made on the reviews, and the rule 2 can also be treated as *review >= 0.8 and avg. reviews >= 4.5*.The result after this pre-processing is listed below:

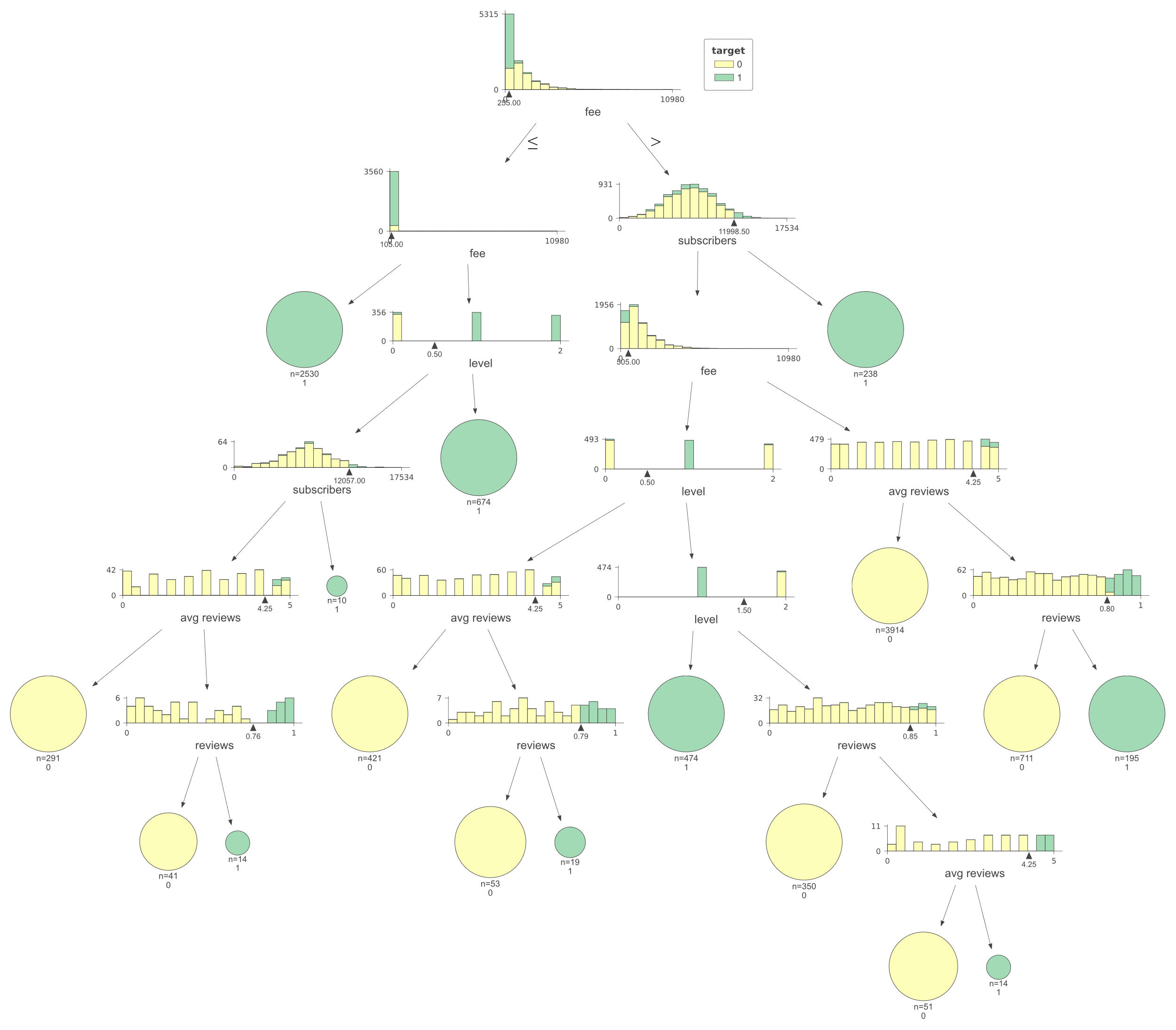


Figure 6: Decision Tree generated with conversion of reviews

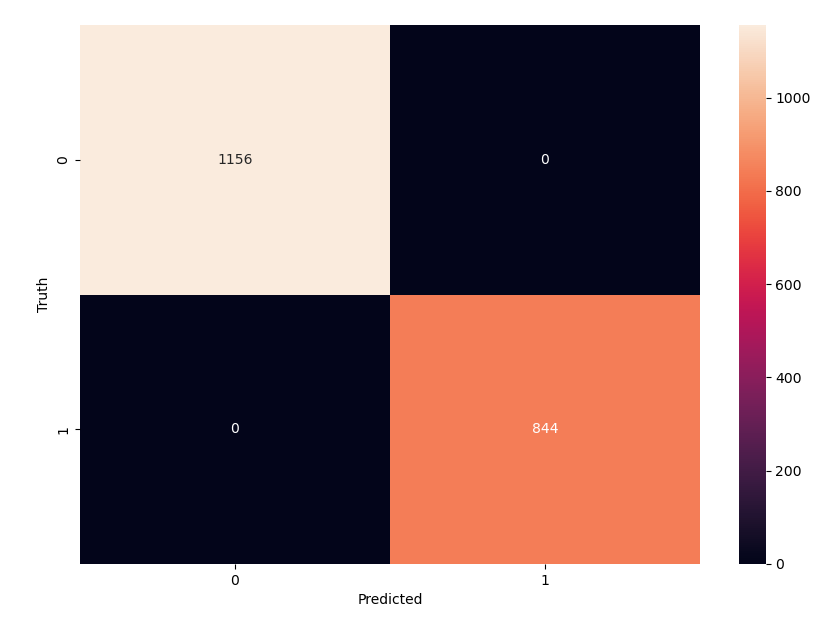


Figure 7: Confusion matrix of decision tree with conversion of reviews

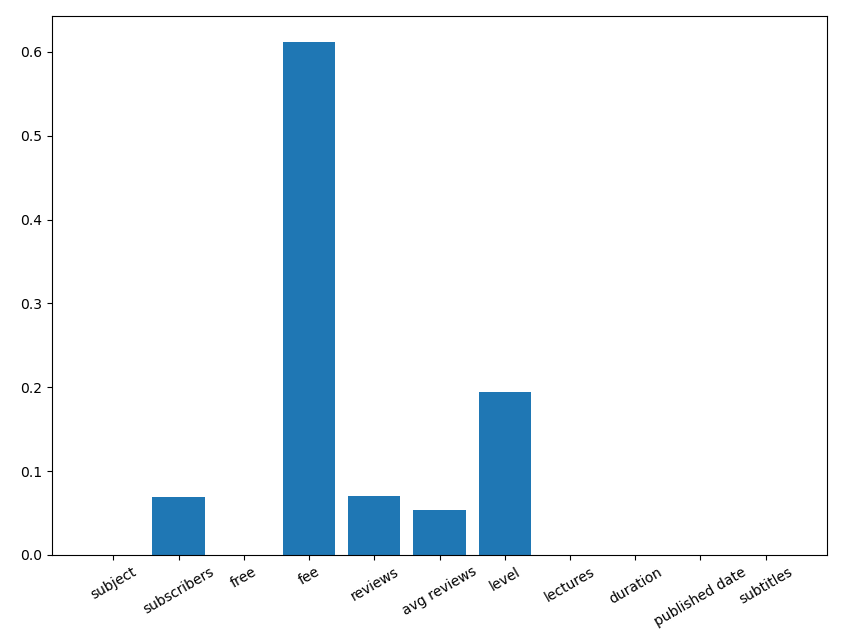


Figure 8: Feature importance of decision tree with conversion of reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 1.00 | 1.00 | 1.00 | 1156 |
| 1 | 1.00 | 1.00 | 1.00 | 844 |
| accuracy |  |  | 1.00 | 2000 |
| macro avg | 1.00 | 1.00 | 1.00 | 2000 |
| weighted avg | 1.00 | 1.00 | 1.00 | 2000 |

Table 2: Classification Report of Decision Tree with conversion of reviews

According to figure 6, the rule generated by the classifiers are:

|--- fee <= 255.00

| |--- fee <= 105.00

| | |--- class: 1

| |--- fee > 105.00

| | |--- level <= 0.50

| | | |--- subscribers <= 12057.00

| | | | |--- avg reviews <= 4.25

| | | | | |--- class: 0

| | | | |--- avg reviews > 4.25

| | | | | |--- reviews <= 0.76

| | | | | | |--- class: 0

| | | | | |--- reviews > 0.76

| | | | | | |--- class: 1

| | | |--- subscribers > 12057.00

| | | | |--- class: 1

| | |--- level > 0.50

| | | |--- class: 1

|--- fee > 255.00

| |--- subscribers <= 11998.50

| | |--- fee <= 505.00

| | | |--- level <= 0.50

| | | | |--- avg reviews <= 4.25

| | | | | |--- class: 0

| | | | |--- avg reviews > 4.25

| | | | | |--- reviews <= 0.79

| | | | | | |--- class: 0

| | | | | |--- reviews > 0.79

| | | | | | |--- class: 1

| | | |--- level > 0.50

| | | | |--- level <= 1.50

| | | | | |--- class: 1

| | | | |--- level > 1.50

| | | | | |--- reviews <= 0.85

| | | | | | |--- class: 0

| | | | | |--- reviews > 0.85

| | | | | | |--- avg reviews <= 4.25

| | | | | | | |--- class: 0

| | | | | | |--- avg reviews > 4.25

| | | | | | | |--- class: 1

| | |--- fee > 505.00

| | | |--- avg reviews <= 4.25

| | | | |--- class: 0

| | | |--- avg reviews > 4.25

| | | | |--- reviews <= 0.80

| | | | | |--- class: 0

| | | | |--- reviews > 0.80

| | | | | |--- class: 1

| |--- subscribers > 11998.50

| | |--- class: 1

The rule is also log in *decision\_tree\_with\_conversion.log*. After compressing the rule, and prune the unnecessary rule (or rewrite in a more simple way), it will become:

1. fee ⩽ 105 ⟶ recommend **⟹ Correspond combination of rules 3,4,and 5**
2. avg. review > 4.25 ⟶ reviews >0.81 ⟶ recommend **⟹ Correspond rule 2**
3. subscribers > 11998.50 ⟶ recommend **⟹ Correspond rule 1**
4. 255 ⩾ fee > 105 ⟶ level = intermediate or expert ⟶ recommend **⟹ Correspond combination of rules 4,and 5**
5. 505 ⩾ fee > 255.00 ⟶ level = expert ⟶ recommend **⟹ Correspond rule 5**
6. Others not recommend

Generally, the rules generated by decision tree classifiers is similar to the absolutely-right rules designed. Besides, the feature importance plot also supports this inference as all the important features are all included in the absolutely-right rules designed. Thus, in decision tree classifiers, how the original rule designed will affect the performance, or in other words, how the features are pre-processed can affected the performance.

### KNN

Now let’s look at KNN. From the observation above, data with conversion is directly be used. Besides, normalization is done as it really affects the performance of KNN, because in KNN algorithms, the distance calculation involved.

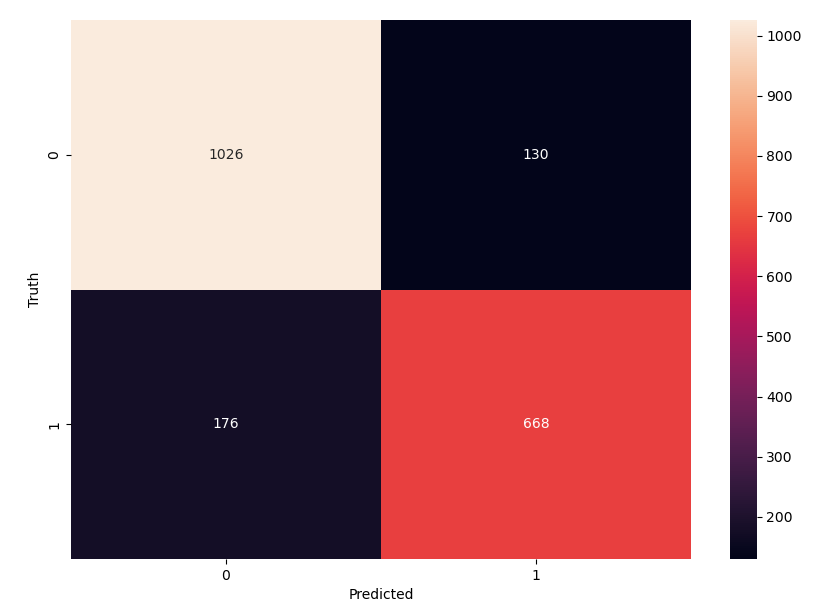


Figure 9: Confusion matrix of KNN(n\_neighbors = 8) with conversion of reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.85 | 0.89 | 0.87 | 1156 |
| 1 | 0.84 | 0.79 | 0.81 | 844 |
| accuracy |  |  | 0.85 | 2000 |
| macro avg | 0.85 | 0.84 | 0.84 | 2000 |
| weighted avg | 0.85 | 0.85 | 0.85 | 2000 |

Table 3: Classification Report of KNN(n\_neighbors = 8) with conversion of reviews

The following is the number of each wrong cases:

1. 'Should not recommend': 130
2. 'Not recommend subscriber > 12000': 29
3. 'Not recommend review >= 0.8\*subscriber && avg.reviews >= 4.5': 56
4. 'Not recommend level = beginner and fee <= 100': 2
5. 'Not recommend level = intermediate and fee <= 250': 9
6. 'Not recommend level = expert and fee <= 500': 80

This is the data when setting n\_neighbors as 8. But when set n\_neighbors=10

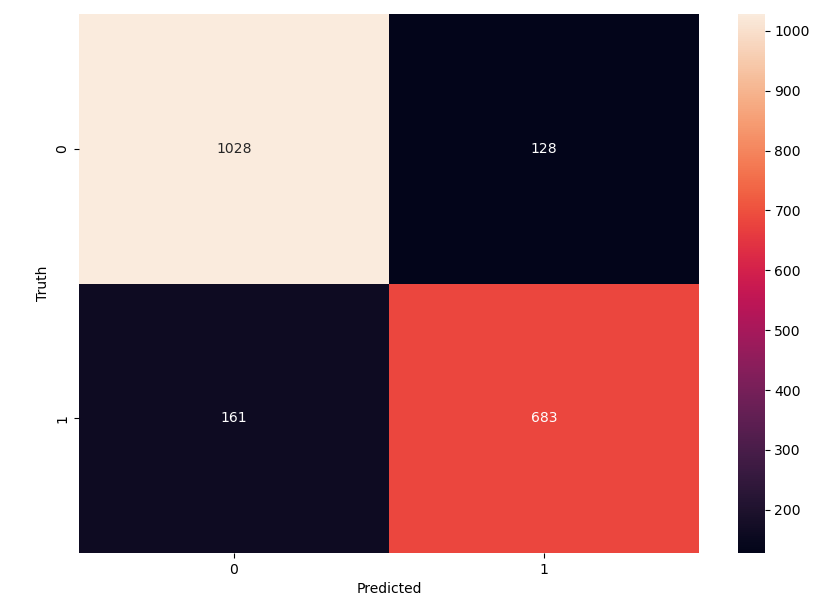


Figure 10: Confusion matrix of KNN(n\_neighbors = 10) with conversion of reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.86 | 0.89 | 0.88 | 1156 |
| 1 | 0.84 | 0.81 | 0.83 | 844 |
| accuracy |  |  | 0.86 | 2000 |
| macro avg | 0.85 | 0.85 | 0.85 | 2000 |
| weighted avg | 0.86 | 0.85 | 0.86 | 2000 |

Table 4: Classification Report of KNN(n\_neighbors = 10) with conversion of reviews

The following is the number of each wrong cases:

1. 'Should not recommend': 128
2. 'Not recommend subscriber > 12000': 28
3. 'Not recommend review >= 0.8\*subscriber && avg.reviews >= 4.5': 54
4. 'Not recommend level = intermediate and fee <= 250': 9
5. 'Not recommend level = expert and fee <= 500': 70

Obviously, increasing n\_neighbors can improve the performance, but only slightly. It looks like it cannot reach the performance like decision tree, as the rules are designed for decision tree.

### Naïve Bayes

Normalization is also be done as there it is better for discretize the range into bins.

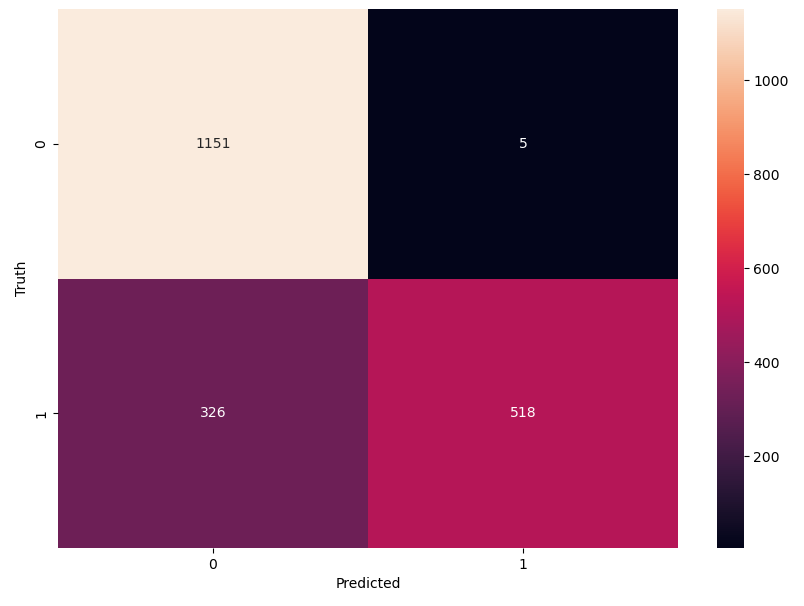


Figure 11: Confusion matrix of Naïve Bayes with conversion of reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.78 | 1.00 | 0.87 | 1156 |
| 1 | 0.99 | 0.61 | 0.76 | 844 |
| accuracy |  |  | 0.83 | 2000 |
| macro avg | 0.88 | 0.80 | 0.82 | 2000 |
| weighted avg | 0.87 | 0.83 | 0.83 | 2000 |

Table 5: Classification Report of Naïve Bayes with conversion of reviews

The following is the number of each wrong cases:

1. 'Should not recommend': 5
2. 'Not recommend subscriber > 12000': 47
3. 'Not recommend review >= 0.8\*subscriber && avg.reviews >= 4.5': 67
4. 'Not recommend level = beginner and fee <= 100': 2
5. 'Not recommend level = intermediate and fee <= 250': 49
6. 'Not recommend level = expert and fee <= 500': 161

By observation, Naïve Bayes has a good precision on recommend courses. Actually that is what a recommender system expected to have, i.e., recommend those good courses but don’t recommend too many bad courses. Thus, even though the recall is low, Naïve Bayes classifiers can still be considered on this kind of task. Naïve Bayes doesn’t reach the performance of decision tree because Naïve Bayes prefer dataset where each feature is independent with other, but this dataset isn’t.