Youth Recidivism Final Project Report

Wei Luo

Goal

As the Project description mentions, the initial goal of this project was to examine the relationship between YES rate, the demographic features of youth and the caseworkers’ note.

Before Project Deliverable 1, I accomplished the Topic Modeling of caseworkers’ note and did some preliminary analysis on the result of Topic Modeling.

By far, all three questions in project description has been answered. The attributes of the young people’s lives are already summarized, so as the outcomes of young people, which was done in the previous group. The role of relationships with DYS staff play in a young person’s decisions to engage with YES has been examined, while there is by far no obvious relationship between these two taken into consideration of the caseworker’s notes.

Data Collection

This step was done by our partners (Codebook\_Case Management Notes.xlsx and DYS\_Casenotes\_16Oct20\_merged\_final\_bydischarge.xlsx).

Preprocessing

In this step, after we talk to the clients, we divide the records into 4 groups by WHO\_WAS\_CONTACTED: Youth only, Staffing, Collaterals, and Youth with others present, and label them 1,2,3 and 4. (Decision Tree has another preprocessing step, which is explained later).

Topic Modeling Analysis

Step 1: Clean text

I started cleaning text by making all letter lower case, tokenize, lemmatize and remove all stop words that I previously defined, which returns a list of words in the original sentence.

Step 2: Make Bigram and Trigram of data

Bigram is the technique that looks at the combination of neighbor two words in a text, e.g. in a sentence: “I am happy”, bigram will look at “I am” and “am happy” in this case, and trigram by its name will look at the combination of neighbor three words.

Step 3: Create Corpus and build model

By using lemmatization, I create a corpus that contains only nouns, adjectives, and adverbs in the dataset. After finish building corpus, I build a LDA model using current corpus and group the caseworkers’ notes by 5 groups, which I will treat as ‘label’ in later analysis. I save this model and corpus to ‘lda.model’ and ‘corpus.txt’. Later I use PyLDAvis to generate a topic modeling visualization result. Below is a screenshot of visualization.

Chart, bubble chart

Description automatically generated

Step 4: Decision Tree Analysis on the grouping result and modification to previous steps

Chart, line chart

Description automatically generated I want to see the relationship between LDA grouping results and current features that I have. After trying different combinations of features on decision tree accuracy score, I find out that contact type, commit days, ’WHO\_WAS\_CONTACTED’ and ‘WHY’ columns are more relevant to the grouping results, which have an 61% accuracy on whole dataset. As I see that using one hot encoder will be more suitable for WHY and contact type, both of which consists of different combinations of categorical features, I apply a one hot encoder to both of them and generate a new dataset **(full\_DTdata.csv)** and train them with decision tree. However, the Decision Tree has depth of 66 and number of leaves is 37161, which clearly means that no obvious relationships (big trends) can be find between features and labels **(See DecisionTree\_commitdays, contact type, why.pdf**, this pdf has a depth of 10 as it is maximum that graphical result can be generated). The only way to reduce depth is to reduce the dimension of dataset. After that, I apply a PCA on the dataset, trying to reduce the dimension, but only find that the feature columns’ slope is never horizontal until the very end, meaning that no feature can be dropped. (On the above is the PCA’s graph).

Then I tried to study on the effectiveness of alpha in decision tree and its relationship with accuracy score, below is the result:

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

From the above two graphs, it is clear that with increasing alpha, accuracy is decreasing, and this method fails because the default alpha for decision tree is 0 and increasing alpha will only decrease accuracy by huge amount. Then I can conclude that all the features are necessary and that there is no obvious relationship between contact type, commit days, ’WHO\_WAS\_CONTACTED’, ‘WHY’ feature and our grouping results.

Diagram

Description automatically generated After talking to our client, I only study on relationship between our grouping result and ‘YES reengagement rate’ and apply the technique ‘keyword extraction’ on input texts of LDA model. The package that I use to realize this technique is ‘Jieba’, which has a keyword extraction function extract-tags that allow us to extract key words, then we can ignore bigram as it is no longer meaningful. New grouping result is in lda3.model and corpus3.txt, visualization is in lda3.html. I first run grouping result as label, only to find a decision tree with depth 1, which did not predict YES or no as classification is evenly distributed (value at left node is almost the same as value at right node). Below is the result:

Diagram

Description automatically generatedThen I tried using grouping result as feature and YES reengagement as label, only the find the same result, where each leaf node does not lean toward any group (see below):

Then I can finally conclude that using LDA model’s result does not associate with YES reengagement rate.

Statistical Analysis on Caseworkers notes

As LDA model fails to find out what causes youth to reengage in YES program, we apply a statistical analysis on the dataset.

Step 1: case note length analysis

In this analysis, we analyze the length of case note on both groups. And the results show that not too many differences are found, while the mean and standard deviation of two groups are close.

Step 2: words choice analysis

In the first step, we group the caseworkers notes into two groups: notes that also have a positive on YES column, and notes that have a negative on YES column, and then we use a dictionary to analyze the word frequencies.

Text, letter

Description automatically generated

In the above screenshot, we picked the 50 most common words among notes that have a positive on YES reengagement and notes that have a negative on YES reengagement, where each dictionary consists of a word-frequency pairs. (top is positive YES and bottom is negative YES.)

A picture containing text

Description automatically generatedThen we use set differences to find out the differences between positive YES reengagement and negative YES reengagement. (below is the result)

Now we can see that not too many differences are displayed, as both groups have similar word choices. And this is probably the reason that leads to the failure topic modeling as topic modeling is also grouping on different words.

While statistically nothing has been found too much different between two groups, I noticed that in positive group, it consists of words such as ‘good’ that imply a positive sentiment while negative group does not have, which leads us to a further sentiment prediction analysis of dataset.

Step 3: Create Word Cloud by caseworkers’ success rate

In this part of analysis, we group by caseworkers and sort them by success rate. I divide 125 caseworkers into two groups: top 60 and bottom 60, and I generate Word Cloud from two groups’ texts. (All of them available in Analysis\_of\_best\_vs\_worst.ipynb)

Sentiment Analysis on Caseworkers notes

Since the dataset does not consist of sentiment, or ratings, we import a pretrained model of sentiment prediction and use that to predict each text’s sentiment.

In this section, we applied package Flair to our dataset. We first groupby MID and append all the texts that belongs to the specific youth, as for the same target, the yes\_reengagement is the same, we predict sentiment of each documents that is grouped by same person with Flair and generate a csv file(sentiment\_analysis\_result.csv).

From confusion matrix analysis, assuming that positive sentiment = YES reengagement, and negative sentiment != YES reengagement, misclassified data are extremely high(this assignment only accounts for less than 30% of dataset) and if we flip relation around, positive sentiment != YES reengagement, and negative sentiment = YES reengagement, it will accounts for around 66% of data, which seems higher, but not reasonable.

Therefore, we utilized another package NLTK to validate this result. However, with NLTK pretrained model, nearly all sentiment analysis result leads to neutral, meaning that above two packages fail to link sentiment case notes to yes engagement rate.

From my reasoning, the major reason these two sentiment analysis tools will fail is that NLTK found nearly all texts to be neutral, and most texts involve with mostly information other than sentiment, and Flair package evaluate most of the hidden sentiment under texts to be negative, as information revealed by texts are mostly engaged with criminal behaviors or events that causes the package to evaluate everything as negative. And it does not mean that there is not a relationship between sentiment and YES reengagement rate. It is just the packages that we use are not good enough.

Conclusion

As from previous three analysis, Topic Modeling Analysis, Statistical Analysis and Sentiment Analysis, we cannot see definite relationships between case notes and YES reengagement rate.

Future Possible Analysis on Dataset

Future Planning Analysis

As sentiment analysis and topic modeling analysis are seemingly not suitable for this dataset. We decided to put the future emphasis on Future Planning Analysis.

Future Planning Analysis is similar to sentiment analysis, while future planning analysis focus on whether the conversion include future planning or not. In order to realize this analysis, we will first choose 50 most relevant words to future planning in the corpus and then either apply a pretrained word2vec model to every word and check if similar words to future planning are obtained, or we will build the word2vec model on our own by manually improve the dataset and train the part that we improve with word2vec model and then apply a semi-supervised model to predict the rest of dataset.

Once we get all related words as a corpus of future planning, we can apply a PCA on all related words feature analysis, and model building will be straightforward after this step.

(I have included a startup file on Future Planning Analysis)

Direct Decision Tree Analysis

Chart, line chart

Description automatically generatedAs everything for this semester is done by analyzing the case notes, when I was trying to reproduce the results, I found that may be the analysis of case notes is not definitely need. As I included contact type, commit days, ’WHO\_WAS\_CONTACTED’ and ‘WHY’ columns that were previously used to analysis Topic Modeling, and use those four columns to directly predict YES engagement rate, I found the accuracy of decision Tree algorithm to be surprisingly high (96% accuracy when maximum depth is not restricted). Tree depth in this case is although 57, the number of leaves is significantly lower (around 5000), which means that at least some trends is showing up. Therefore, I further did some experimenting on this result, on the right is the depth vs accuracy plot. Therefore, if the maximum depth is around 20, we can actually achieve an accuracy of 90%. I have actually included summary results of branches and total instances under this branch in

**Future\_analysis\_DT\_summary.csv**

In this file, we can see that decision tree algorithm indeed find trends in this case as the maximum instance of a branch is around 10,000. If time allow, I will study decision paths on each trend to find out which combinations of decisions is impacting on YES reengagement rate.

Appendix

Guidance to reproduce results

In the section, I will walk you through the code section of final report. All the dataset and some of the inputs to the model is in this folder.

1. Preprocessing of dataset is in preprocessing/general\_preprocessing.ipynb

Result if general\_preprocessing is in Dataset/train.pickle

1. Then we continue to LDA\_model\_generator and inside, we have two model generators:

LDA\_with\_nltk.ipynb and LDA\_with\_Jieba.ipynb

Result of LDA\_with\_nltk.ipynb is in LDA\_Output/lda.html

Result of LDA\_with\_Jieba.ipynb is in LDA\_Output/lda3.html

(Note: everything after this step will use result of LDA\_with\_nltk.ipynb, as LDA\_with\_Jieba.ipynb grouping result is less accurate)

1. The preprocessing step of DecisionTree is in preprocessing/ DecisionTree\_preprocessing.ipynb.

Result of this preprocessing is full\_DTdata.csv in Dataset folder.

1. Then LDA’s step 4 decision tree algorithms is in DecisionTree\_Analysis/ DecisionTree\_LDA\_analysis.ipynb

All Results of DecisionTree\_LDA\_analysis.ipynb are in DecisionTree\_Output.

Including:

DecisionTree’s result of 4 selected columns: contact type, commit days, ’WHO\_WAS\_CONTACTED’ and ‘WHY’ **vs** LDA’s grouping result.

DecisionTree’s result of grouping result **vs** YES

DecisionTree’s result of YES **vs** grouping result

1. Then we did statistical analysis and sentiment analysis, both of them can be found in their separate folders. (Result of these two analysis are all in ipynb files)
2. Finally, Future\_Analysis part include Word2vec\_analysis.ipynb as startup, and Decision\_Tree\_Analysis\_on\_Other\_Featues.ipynb. The result of this Decision Tree is also in DecisionTree\_output/ Future\_Analysis\_DecisionTree\_result.dot.pdf

and DecisionTree\_output/ Future\_analysis\_DT\_summary.csv

You can follow the code in the file.

NOTE: reproduction of codes is based on the fact that you have copied and paste the whole directory and that you are using MAC system. And if there exists error FileNotFound, please check all the paths.