Project Deliverable 3

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 Aggregation

This step was done by previous group (see Codebook\_Case Management Notes.xlsx)

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(see lda.html). Below is a screenshot of visualization.

Chart, bubble chart

Description automatically generated

Step 4: Decision Tree Analysis on the grouping result

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 80% 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 **(DTdata.csv)** and train them with decision tree. However, the Decision Tree has depth of 69, 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 right 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

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.

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, text

Description automatically generated

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 no 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.