Université de Montréal

Job dissatisfaction detection through progress note

par

Jiechen Wu

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Discipline

17 novembre 2021

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Job dissatisfaction detection through progress note

présenté par

Jiechen Wu

a été évalué par un jury composé des personnes suivantes :

Jian-Yun Nie
(président-rapporteur)

Philippe Langlais
(directeur de recherche)

Nadia Lahrichi
(codirectrice)

Guy Lapalme
(membre du jury)

Résumé

La détection d'insatisfaction basée sur les notes de progression rédigées par des soignants de la santé domestique attire de plus en plus d'attention en tant que méthode de sondage, ce qui aidera à réduire le taux de rotation du personnel soignant. Nous proposons d'étudier la détection d'insatisfaction du soignant comme un problème de classification binaire (le soignant est susceptible de quitter ou pas).

Dans ce mémoire, les données réelles de six mois recueillies à partir de deux agences de soins à domicile sont utilisées. Après avoir montré la nature des données et le prétraitement des données, trois tâches de classification avec des granularités d'échantillonnage différentes (par note, par période et par soignant) sont conçues et abordées. Différentes combinaisons d'hyper-paramètres d'étiquetage sont soigneusement testées. Différentes méthodes de découpage sont couvertes pour montrer les limites des performances théoriques des modèles. L'aire sous la courbe ROC est utilisée pour évaluer les limites des approches mises en place que nous aurons mis en place. Les 6 ensembles d'attributs textuels et statistiques sont comparées. Enfin, les caractéristiques importantes des résultats sont analysées manuellement et automatiquement.

Nous montrons que les modèles fonctionnent mieux "par note" et "par période" que "par soignant" en termes de classification des notes. L'analyse manuelle montre que les modèles capturent les facteurs d'insatisfaction bien qu'il y en ait assez peu. L'analyse automatique n'exprime cependant aucune information utile.

Mots-clés: Détection d'insatisfaction, santé à la maison, rotation, note de progression, fouille de texts, classification de texte.

Abstract

Dissatisfaction detection based on the home health caregiver's progress note draws more and more attention as a probing method, which will help lower down the turnover rate. We propose to study the detection of dissatisfaction of health caregiver as a binary classification problem (the caregiver is likely to "leave" or "stay").

In this master thesis, the real six-month data collected from two home care agencies are used. After showing the nature of the data and the prepossessing of data, three classification tasks with different sample granularity (note wise, period wise and employee wise) are designed and tackled. Different combinations of labeling hyper-parameters are tested thoroughly. Different split methods are covered to show the theoretical performance boundaries of the models. The under the ROC curve area (AUC) scores are reported to show the description ability of each model. The 6 sets of textual and statistical features' performance are compared. Lastly, the important features from the results are analyzed manually and automatically.

We show that models work better on note wise and period wise than employee wise in terms of classifying the notes. The result of manual analysis shows the models capture the dissatisfaction factors, although there are quite few. The result of automatic analysis doesn't show any useful information.

Keywords: dissatisfaction detection, home health, turnover, progress note, text mining, text classification.

Contents

List of tables

List of figures

List of symbles and abreviations

NLP	Natrual Language Processing
ROC	Receiver operating characteristic
AUC	Area under the ROC Curve
PCA	Principal Component Analysis
TF-IDF	Term frequency-inverse document frequency
LM	Language Model

Acknowledgments

From the knowledge embedding research to this dissatisfaction detection task, I enjoyed the every moment that I was working in this field. Coming back to the campus again 10 years after graduation in a grateful manner, I feel my brain is sharpened again and my passion is sparked.

First of all of course, I want to thank my director Philippe Langlais, who helped me find the right NLP tool and did thorough editing work on my thesis. Then I want to thank my co-director, Nadia Lahrichi, for her advice and comments, and all the people at the RALI who helped me recognize Philippe's hand writing comments.

At last, I want to thank my friend Anna who helped me do the proofreading and encouraged me lot, and my wife Shanshan who always supports me and has the faith in me.

Introduction

During the COVID-19 pandemic, Quebec has been one of the most affected provinces in Canada. Home care staffs play a key role to reduce pressure on the hospital system in the era of COVID-19. By 2050, people over the age of 60 are expected to be doubled [42]. It asks the home healthcare industry to hire and retain high quality home care workers to meet growing demands.

Among the issues raised, staff turnover is a major problem in the home healthcare industry. The caregiver turnover/churn rate[1] has reached 65.2% in 2020 [34]. Limited caregivers and high turnover rates can bring a diminishing effect on healthcare infrastructure and be very challenging.

In a context where the work of the caregivers is particularly based on a relationship of trust with the patients, the prevention of turnover is crucial for the continuum and the quality of the care provided. Losing a caregiver worker therefore has negative impact for all the actors involved: the quality of care is affected, the schedules are upset, and it may even happen that some visits are cancelled due to a lack of staff.

In this research, we collaborated with an industrial partner, AlayaCare, specializing in the design of integrated software for home care. As its survey revealed [16], 53% of the respondents believe the caregiver shortage is the main barrier in their business. About 60% home care agencies agree that COVID-19 makes the situation worse for most of them [17].

Some popular economic solutions are motivating employees with raises, benefits and other compensations, but this might not be the silver bullet. Zeytinoglu [44] found only the 23.7% and 10.3% personal support workers left agencies due to the dissatisfaction of the pay and benefits, respectively. However, there is a much larger degree of dissatisfaction about other aspects, such as: the lack of support from supervisors (18.6%), lack of support from co-workers (7.2%), lack of job security (13.4%), work-related stress is (12.4%).

Conventionally, agencies use some "passive" methods to collect and diagnose the dissatisfaction, e.g.: survey, in-person communication, etc. It means the agencies have to passively rely on the caregivers to report their dissatisfaction. We suggest an "active" method to detect them based on narrative progress notes by using natural language processing (NLP)

[1]Turnover rate is the percentage of employees that leave during a certain period of time.

algorithms. During each visit, caregivers are supposed to note their work content, patients' clinical status and achievements. Some of them will type the notes into the AlayaCare system by themselves, others will get help from the coordinates for the input work. Notes are either in English or French, which depends on the different agencies. In our research, we work on the English notes solely. The idea of detection of the dissatisfaction from narrative text is not new. Researchers reported their dissatisfaction detection work by extracting the info from product reviews, surveys, social media posts etc. However, there's no related work been done with progress note yet.

The aim of this research is to develop a tool helping to diagnose employee dissatisfaction based on the progress notes. This tool will allow home healthcare agencies to be proactive in managing staff turnover. It will allow them to identify the irritants in the daily work of the staff, as well as the causes that may lead to resignations. It will then be directly used in planning the work of employees with the aim of providing a work environment that better meets expectations.

The questions we try to answer are :

- Can we detect the caregivers' dissatisfaction from the notes?
- What are the factors/indicators for the detection in the note?
- Are these indicators causes or just results?

We show that classifier we trained is able, under some assumptions, to correctly classify the note. It may even be used to deliver important indicators.

This thesis begins by presenting related work. Following this, it explains the corpus,and the data cleaning process we conducted. It then presents the assumptions, the data labelling, three sub-tasks and the evaluation. Subsequently, it explores the classification performance of our experiments and the dominant features.

Chapter 1

Literature Review

Global healthcare systems have been impacted by the aging population. This environment is experiencing a shortage of skilled healthcare workforce [9, 28]. Moreover, the average turnover rate in home care sector is about double or triple the rate in other healthcare sectors [44]. Amid the COVID-19, the psychological distress and the turnover intentions have increased [24]. The home care agencies have to ensure an attractive environment. The impact of employee dissatisfaction is multiple: on the health of the caregivers themselves but also on the quality of services offered to patients [40, 27, 30]. Detecting caregiver's dissatisfaction is a critical piece in the puzzle which helps to understand and resolve the caregiver turnover problem.

Qualitative methods and psychometric tools are used in the majority of work in the literature. There is the work of assessing the relationship between employee dissatisfaction and their response to this dissatisfaction [41] and on the reasons for dissatisfaction [8]. The satisfaction rate of healthcare worker in hospitals in the United States are often measured by Satisfaction of Employees in Health Care (SEHC) survey. The model covers some elements, such like: the workplace experience, the communication with supervisors and managers, and the relationship with colleagues. These elements are very similar to those used in industry in general.

The employee's voice can be heard not only in the surveys, but also in the posts on social media. Goldberg and Zaman [5], for example, show how textual analysis of indeed.com reviews is useful to identify the most pressing issues of employee dissatisfaction. The online employee reviews posted on jobplanet.co.kr are used in the work of Jung and Suh [21] to identify job satisfaction factors. Data captured on Twitter [15] have been analyzed to show the effect of the compensation on job satisfaction.

Although the avenues provided by these researches are promising for probing the dissatisfaction of home care employees, the work environment of workers is very different in

form and nature. First of all, many employees arrange their own schedules, so their punctuality is not related to their work attitude. In addition, they sometimes arrange the case load according to their needs to achieve the work-life balance. The supervisor-supervisee relationship therefore remains limited. Likewise, they have few meetings and less interaction with colleagues.

Hertz and Lahrichi [1] did the only work which relates to employee satisfaction in a Quebec home care context. This work has a completely different objective, though. It discusses the balance of the workload in the different areas by reviewing the territorial division of a home care organization. The management team shows that this balance is the major source of dissatisfaction. Home care agencies regularly plan the home care routes using the load balance as a criterion [11, 26]. The distance traveled and the number of visits are generally used to measure the load balance.

Many researches detect signs of depression from text [2]. They propose different types of classifiers via the supervised learning based on many linguistic traits. Bag-of-words model is among the popular ones. Researchers are building up a list of specific words [27], or using of the Linguistic Inquiry and Word Count tool [43], which counts the occurrences of words associated with certain categories.

The majority of the depression detection researches are based on social networks such as Twitter [14], Reddit [20] or Facebook [37]. Other common datasets are the forums where participants discuss their symptoms or treatment [20]. The other types of writing where the authors do not wish to reveal voluntarily their symptoms remain relatively unexplored.

Research of dissatisfaction by text mining method in the home care sector is therefore still in the early period. It will help us to better understand the employee dissatisfaction, to determine quantitative criteria and to compare NLP techniques in professional lingual context.

Chapter 2

Nature of Data

AlayaCare is a provider of Cloud-based home Healthcare software. The platform created by AlayaCare helps home care agencies manage the entire client life circle, e.g., record patient information, schedule the visits, assign the tasks, etc. There are some terms that we will use often in this thesis:

- Agency/Tenant: Agencies are the home healthcare companies who buy the access to the cloud based platforms developed by AlayaCare. In AlayaCare, agencies are generally referred as tenants.
- Caregiver/Employee: Caregivers are the healthcare workers who are recruited by the agencies to provide the home healthcare services. In this research, we often use employee to refer to caregiver. Caregivers are the people who write the progress notes.
- Patient/Client: Patients request the home healthcare services from the agencies. Caregivers usually refer their patients as clients. Ct is short for client.

Thanks to the platform that AlayaCare created, when a caregiver from an agency comes to a patient's domicile, caregiver fills the relevant information on the electronic device. Meanwhile, the system will also record the relevant information into the servers. This information includes the structured data (e.g., start date, end date, duration, travel distance, etc.) and unstructured data (e.g., progress note). The structured data are used in another research to detect caregivers' dissatisfaction as well. In our research, we focus on the unstructured data: progress notes. Each tenant keeps its own caregivers and patients. For this reason, all the progress notes from the same tenant are naturally put together in AlayaCare's database. Progress notes of tenant A (data pulling time: 2019-11-21) and progress notes of tenant B (data pulling time: 2020-04-15) are used. In addition, we have been told that crossing the data from two tenants is prohibited, which means we have to treat the two data sources as two separate datasets and train our models on each of them separately.

None of the progress notes is labelled. It's one of the difficulties of this research. Each progress notes is including the info of which employee was involved and when he/she entered the note. Employee status ("active" or "terminated"), which can be seen as the label at employee level, is determined by the duration of the gap between the last note's created time and the data pulling time.

The simplest way to get the note level label is labelling all the "terminated" employees' notes positive and all the "active" employees' notes negative. In this way, we assume that all the "terminated" employees write the notes which will carry the dissatisfaction since their first day of work. It doesn't make much sense.

Generally, we believe that the last notes before the turn over contain some signals of dissatisfaction. For this reason, we only mark the last notes from each "terminated" employee as positive. All the left notes from both "terminated" employees and "active" ones can be seen as negative. If we mark them all as negative and use them, there would be too much negative notes and only a few positive ones. Since the employee's work is scheduled weekly, the amount of weeks naturally become the hyper-parameter to indicate the notes we use in the experiments. In the Chapter Methodology, we will define the hyper-parameters which determine the weeks of positive and negative notes. Meanwhile we will present the values of these hyper-parameters for each experiment.

2.1. Progress Notes

The progress notes, in the AlayaCare system, are the patients' "medical" record notes written by the caregivers after visits. According to AlayaCare's terminology, its tenants' employees (e.g: personal support workers, nurses, personal speech therapist), who provide the health care services, are caregivers[1]. Thus the caregivers generate various kinds of clinical information, during one or more visits, such as: medication taken, wake-up time, bedtime and toilet use time, progress on pronunciation etc.

Fig. 2.1 shows the authors of the 3 kinds of notes in the system. Employee notes and patient notes won't be used in this thesis.

Fig. 2.2 shows the data fields from a progress note file and a couple of demo records. To protect the patients privacy, neither the users' ids nor the note texts demonstrated in this thesis are related to a real case.

Meanwhile, AlayaCare has run a script to anonymize the notes by substituting the people's names. Each name is supposed to be replaced by a place holder $[PEOPLE]$.

[1]Caregivers and employees are used interchangeably.

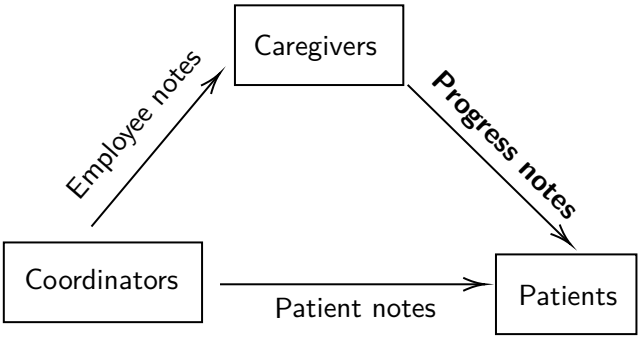


Fig. 2.1. Three kinds of notes and their authors' relationship. The starting point of the arrow indicates the author of that kind of note.

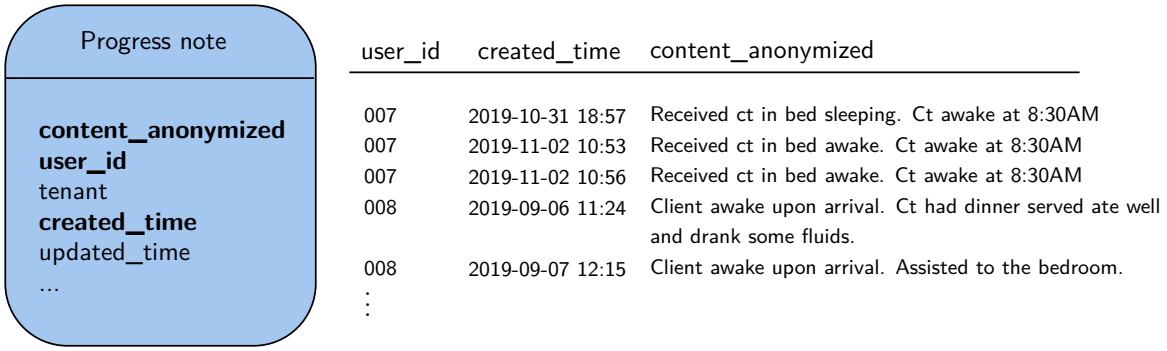


Fig. 2.2. Progress note data fields and part examples. The *user_id* is the employee's unique identifier.

2.1.1. Examples

We present 3 chosen examples of progress notes. Meanwhile, the situations of typo, missing punctuation are kept to show the real data.

Example 2.1.1. *Client opened the door she had late lunch eaten well stay down the whole client son very sick client newspaper we have conversation while playing cards she had a goodday*

Example 2.1.2. *Received ct in the tv room watching tv. Later, writer offered snack but ct said she's fine. Assisted ct to the bedroom after watching tv. Assisted in getting ready for bed. Oral care done.Cbd cream applied on legs with foot massage. C/o burning legs early morning, cold bottles applied. Voided 3x the whole shift no bm. Ct still asleep at end of shift.*

Example 2.1.3. *Received client in his motorized wheelchair. Endorsement done. Medications taken with 2 mugs of juice and soda water. Assisted client to his reclining chair. Personal care done and changed top and pull ups. Escorted to the dining hall for lunch.*

2.1.2. Signals of dissatisfaction

We present 3 examples that show the stress of job, which provide probably the signals of dissatisfaction.

Example 2.1.4. *[...] Mrs A doesn't want me to go inside with her, stayed out few minutes trying to convince her to let me in. I almost had to force myself in when she went inside. Several times tried to persuade Mr A to take a bath and change, no success. From time to time she asked me why I m here and if her husband knows about me... or who let me in. Few times asked me to leave the or she will call someone. [...]*

Example 2.1.5. *[...] Dinner was given to ct but she didn't eat it and became frastuated and tried to through the plate on the ground when we were encouraging her to eat. One of the staff said she loved bread and jam and gave it to ct with milk which ct accepted to eat. Ct completely refused to eat her dinner. [...]*

Example 2.1.6. *No response on arrival at Cl's residence, writer checked with nurse to see if Cl came down for his meds and writer was told Cl hasn't been seen, writer then checked the dinning room and Cl was absent. Writer went back to Cl's residence and knocked but still no answer then writer went back downstairs to check again, Cl was still unseen. [...]*

In Example 2.1.4, the client doesn't refuse caregiver's service; In Example 2.1.5, the caregiver can't get the client to eat by all means; In Example 2.1.6, the client is unseen. In these examples, the caregivers don't express explicitly their dissatisfaction. Nevertheless these stressful situations will affect the employees' job satisfaction.

2.2. Employee status

Employees write the notes of their visits. Each note doesn't necessarily relate to a specific visit. In other words, there's no 1-to-1 mapping between the progress notes and the visits. It adds up to the difficulty of the task of labelling data.

Based on the rules recognized by AlayaCare, we gave all the employees who didn't write a note within the last three months (90 days from the time of pulling data) the "terminated" status, while an employee who did write a note received the "active" status. Fig. 2.3 illustrates the method.

2.3. Data cleaning

Data A and data B are two sets of data coming from two different tenants, respectively tenant A and tenant B. From Table 2.1, we can tell that Data A (110k notes) is about 3 times bigger than Data B (38k notes).

Since we rely on the *created_time* to decide the employee's status as "active" or "terminated", the notes missing *created_time* value are removed in our data cleaning process. Due

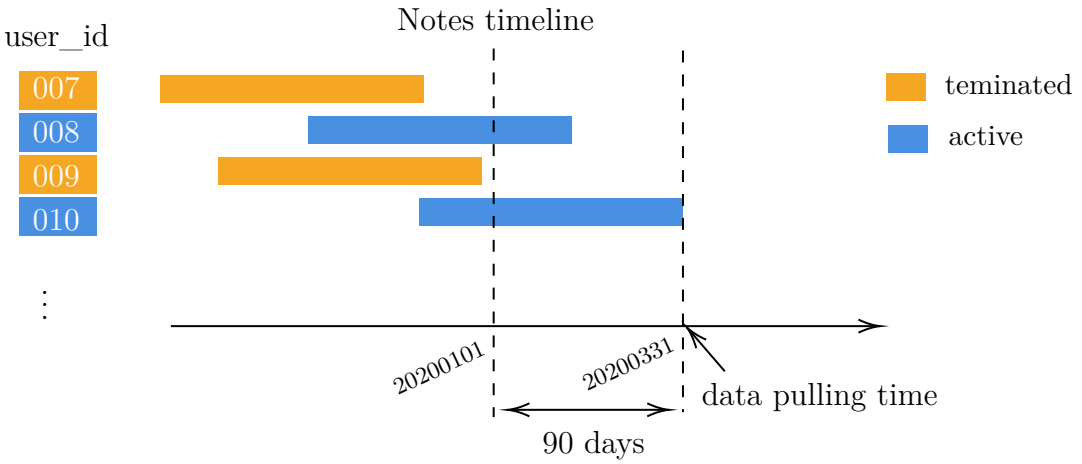


Fig. 2.3. Method for labelling the employee status.

Data A		Data B	
# of notes	# of employees	# of notes	# of employees
110,095	523	37,531	501

Table 2.1. Number of notes and employees

to the fact that AlayaCare was supposed to pull the last 6 months notes of each employee, all the notes of each employee outside his/her last 6 months (184 days) range will thus be cleaned in our prepossessing. As shown in Table 2.2, about 15% of records are cleaned in Data A, meanwhile Data B is not affected at all. The cleaning process didn't affect the number of employees.

Statistics	Data A	Data B
Before cleaning	110,095	37,531
# of notes missing created_time	2,896	0
# of notes outside the last 6 month	17,158	0
After cleaning	90,041	37,531

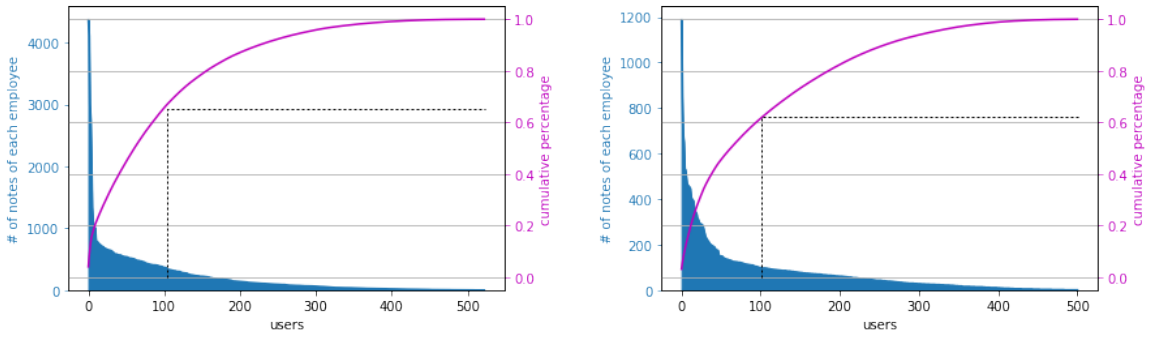
Table 2.2. The types and numbers of invalid records.

2.4. Dataset metadata Profile

As shown in Fig. 2.4, in both Dataset A (90k notes) and Dataset B (38k notes)[2], about 20% of employees wrote about 60% of the notes.

In Fig. 2.5 the weekly total number of notes rises to a relatively high level, which is about 5 times larger than in the past, in both datasets after September 2019. One reason might be

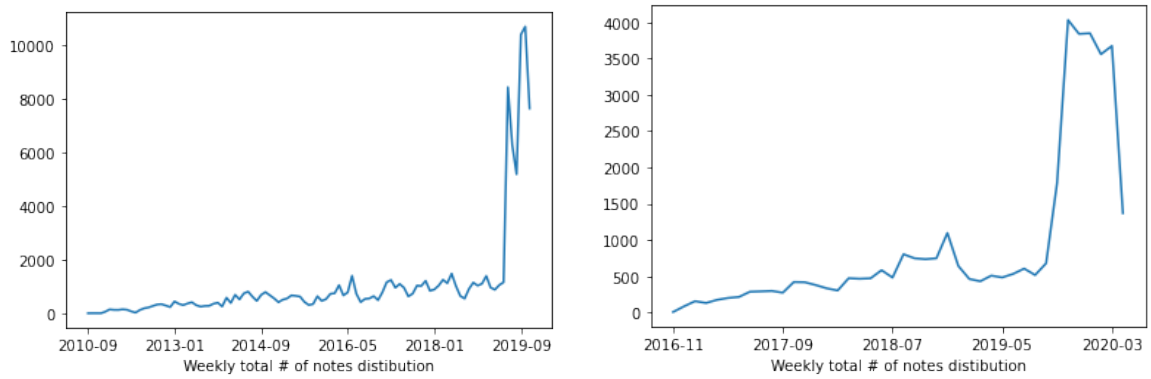
[2]For the convenience, we name the Data A and Data B after data cleaning respectively as Dataset A and Dataset B.



(a) Dataset A (b) Dataset B

Fig. 2.4. Rankings of employees by number of notes

that tenants are modifying their habits in using AlayaCare's tools and are becoming more interested in collecting data about caregivers.



(a) Dataset A (b) Dataset B

Fig. 2.5. weekly total number of notes distribution

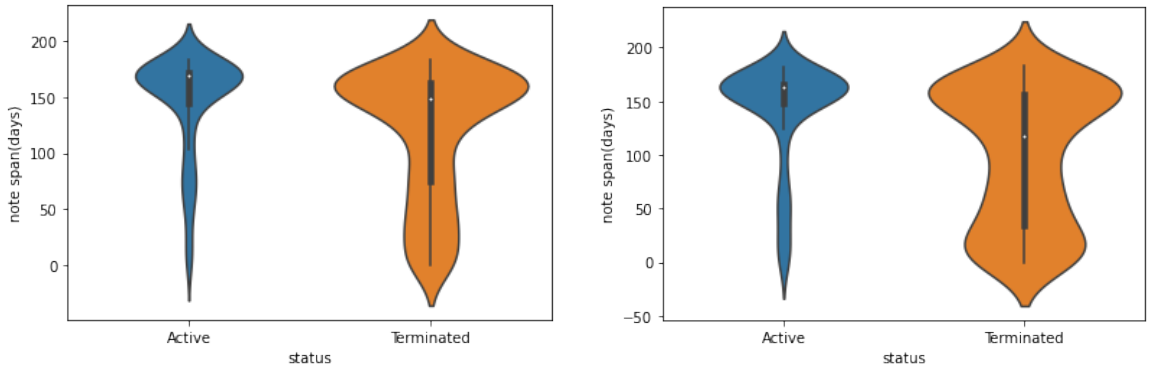
Table 2.3. gives the detailed numbers of records in "terminated" and "active" status in Dataset A and Dataset B. The employees with "active" status wrote more notes than those with "terminated" status in both datasets.

	Dataset A		Dataset B	
Status	# of notes	# of employees	# of notes	# of employees
Terminated	41,096	337	15,969	301
Active	48,945	186	21,562	200
Total	90,041	523	37,531	501

Table 2.3. Number of notes and employees with different status

We define a note time span of an employee as the time duration between his/her first note and last one in our data. The note time span distribution in Fig. 2.6 shows the employees with

the "active" status have commonly longer note time span than those with the "terminated" status.



(a) Dataset A (b) Dataset B

Fig. 2.6. The distribution of the length of note time span. A violin plot is the combination of kernel density estimation and the box plot. It illustrates the distribution meanwhile it shows the median (Q2, indicated by the white spot), first and third quartile (Q1 and Q3, indicated by the lower and upper edge of the thick box), the interquartile range (IQR, indicated the length of the thick box) and the lower and upper extreme (min and max, indicated by the lower and upper edge of the thin box). It shows the median values of time span are for Dataset A around 170 days (active) and around 150 days (terminated); and for Dataset B around 160 days (active) and around 120 days (terminated).

2.5. Normalization of note texts

Some notes are in the HTML format. It seems that the employees may have the freedom to add different styles to the note text with a rich text editor. The style related tags or html special characters include:

- HTML tag (e.g. <p>, ,
)
- HTML entitie (e.g. , &lt, &gt)

Although these HTML tags and entities may indicate personal emotions, we decided to remove them from the text. We also removed all the carriage returns before the tokenization, though there is an idea suggesting that they may as well relate to job dissatisfaction. Finally, the date info in the text is also removed due to possible bias.

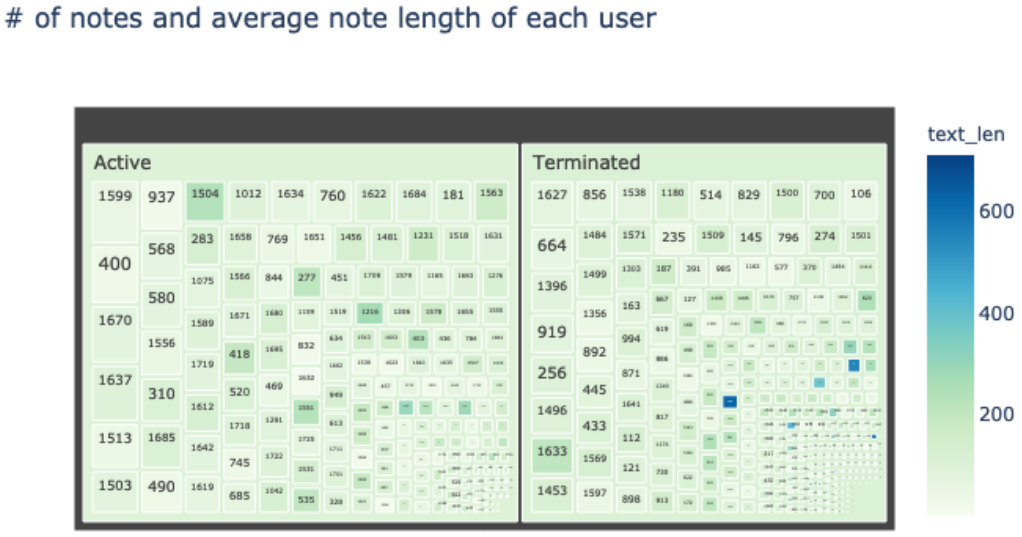
2.6. Length of Note Text

The length of note text, which is the amount of tokenized terms, varies a lot among different records. Some employees tend to write longer notes than others. In the treemap of Fig. 2.7, the size illustrates the number of notes and the color indicates the average length of the notes for each employee and label group.

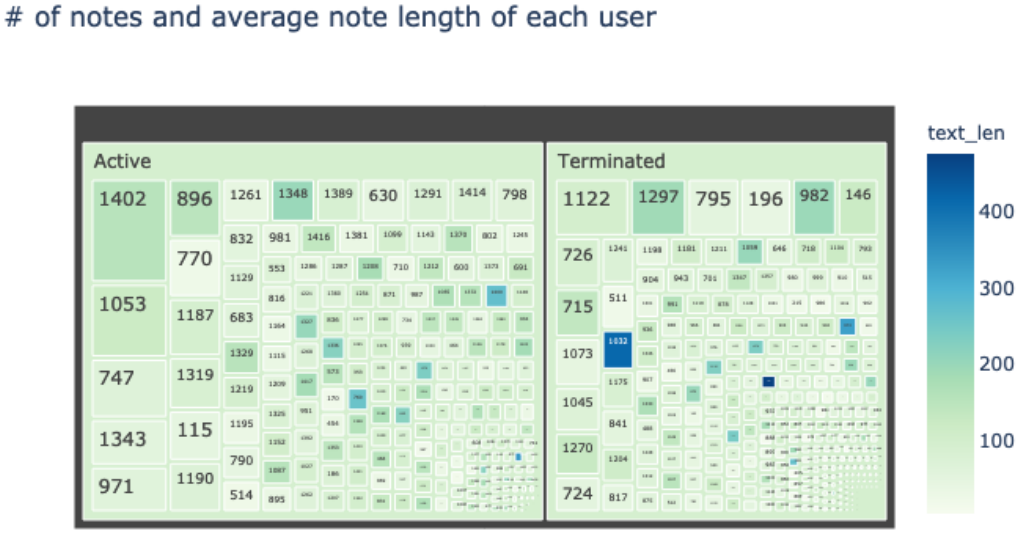
We observe:

- The background color of "active" and "terminated" groups are almost the same, which tells us that active employees wrote on average notes of similar length as terminated employees.
- The size of the "active" group is larger than that of the "terminated", which tells that the active employees wrote on average more notes than terminated employees.
- The background color of most employees are "light green", which tells us that the average text length of most employees is inferior to 200.
- There is no "dark blue" on the "active" employee side, which tells us that those extreme outliers (Dataset A: $text\_len > 400$, Dataset B: $text\_len > 350$) usually show up among the terminated employees.
- The rectangles with different background colors distribute relatively evenly[3], which tells us the average text length distributes relatively evenly among the employees with different numbers of notes.

[3]They are sorted in descending order of area and positioned from left to right, from top to bottom.



(a) Dataset A



(b) Dataset B

Fig. 2.7. The number of notes and average note length of each user. Sorted by the number of notes within each "active" and "terminated" class. In each class, the user with most number of notes is at the top-left corner and the user with least number of notes is at the bottom-right corner.

Chapter 3

Methodology

3.1. Data preparation

As often in a concrete project, data do not come prepared, and we spent great efforts into engineering representative tasks.

3.1.1. Assumptions

The progress note is not a complete record of a patient's key clinical data and medical history. In other words, the employees are not supposed to express explicitly their own emotions nor dissatisfactions in the progress notes. Even so, it shows that the employees are stating the information in a diplomatic and factual manner.

This project is based on a few assumptions.

Assumption 1. The progress notes will provide a signal to predict employees' job dissatisfaction.

There are two kinds of "signals" we expect to detect in the data:

- The changes of language style, which is brought unconsciously into the progress notes by employees when their job dissatisfaction is growing.
- The factors about patient care and work environment, which are considered the direct causes of job dissatisfaction. The factors are: bad outcomes related to patients, insufficient patient response, verbal abuse received, poor cooperation with co-workers, etc. [29]

Assumption 2. The employee's job dissatisfaction will lead to a turnover in the near future.

Without an objective measure of the degree of the job dissatisfaction of each employee, we assume each employee carried a job dissatisfaction by observing his/her termination. We assume that job dissatisfaction and intention to leave are the same. We ignore there are some employees who keep working for a long time ($\geq$ 3 months) while feeling job dissatisfaction.

We assume that the process of getting job dissatisfaction is one directional: As soon as an employee gets it, he/she will never get rid of it.

Assumption 3. The active employees won't stop writing notes for more than 90 days.

With these assumptions, we may separate the employees into two groups "active" and "terminated" by only observing the interval between the date when they wrote last note and the date of pulling date.

3.1.2. Data labelling

Assumption 3 helps us label the employees easily by two classes positive (terminated) and negative (active). In section 2.2 we presented that how can we decide the employee's status. The employee's status "active" or "terminated" are considered as the classes for each employee. Fig. 2.3 illustrates the labelling process.

Since employees don't necessarily carry the job dissatisfaction from their first day of work, we also consider labelling the notes. In other words, we assign each note to a class, either positive (related to terminated) or negative (related to active).

There are two possible boundaries which correspond to two different perspectives of analysis. As shown on the left in Fig. 3.1, from a temporal perspective, we may consider a terminated employee's historical notes as negative data. And we consider those recent notes, whose *created_time* is closer to termination date, as positive. From the tenant level perspective, we label the active employees notes as negative data instead, and the terminated employee's recent notes as positive. Meanwhile, the historical notes are not taken into account.

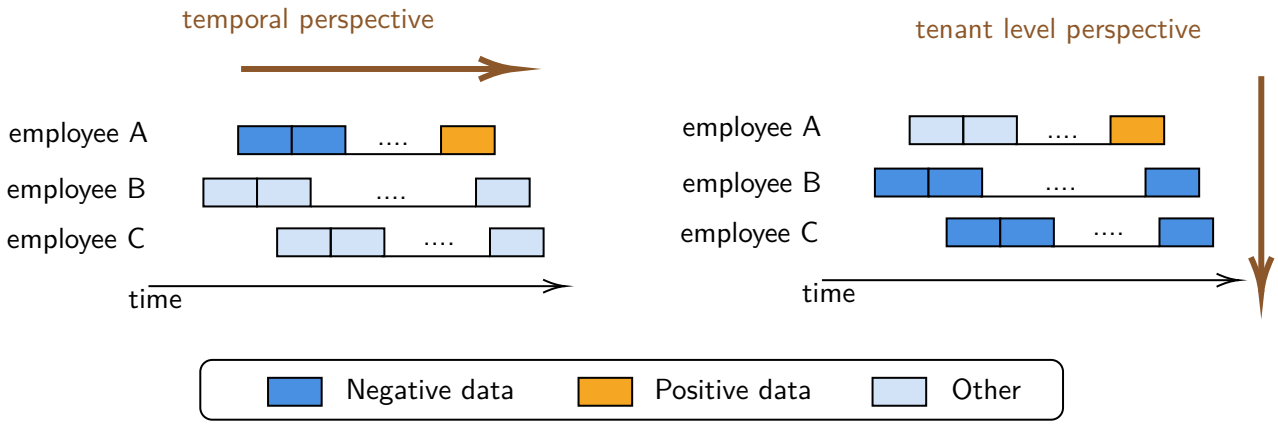


Fig. 3.1. Labelling the data from two different perspectives. Employee A has a terminated status while employees B and C have an active one.

We choose to label this imbalance of the data (many blue only a few orange) as shown in Fig. 3.2 combining these two perspectives. It is simply labelling the terminated employee's recent notes as positive and the other notes as negative.

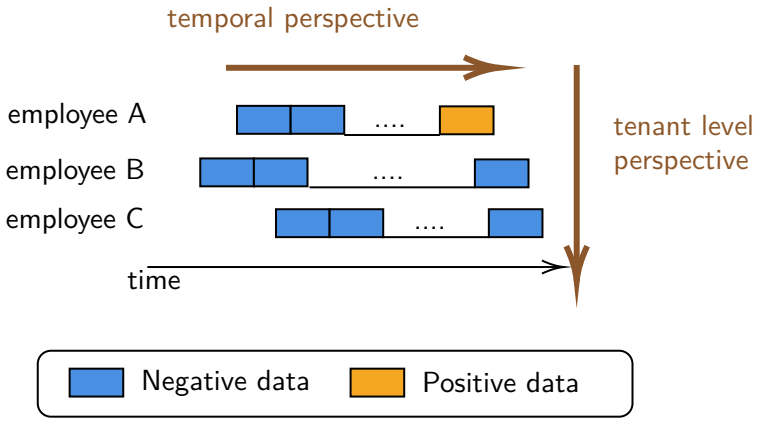


Fig. 3.2. Method used to label the data.

There's probably a transition phase between the negative and positive notes from a terminated employee. We have no way to measure it. We suggest a **safety phase** between them. It is the data that we won't include in the final dataset. In other words, data marked as **safety phase** are not used in the training nor testing. This phase of data are shown as grey rectangles in Fig. 3.3.

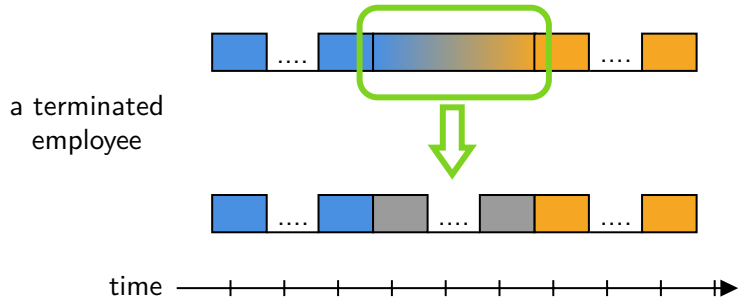


Fig. 3.3. The safety period for a terminated employee.

The recent data, which is close to the data pulling time, of an active employee carries uncertainty. The unseen future data coming after the data pulling time may affect the label of the recent data as shown in Fig. 3.4. There are two possibilities for the unseen data:

- This active employee remains as an active employee.
- This active employee becomes a terminated employee.

For this reason, we will put the active employees' recent data in a **safety phase**, as shown in Fig. 3.5. It means we don't use them in the training nor testing.

There are 3 train/test split related hyperparameters. We align the recent part of a terminated employee's note timeline and an active employee's note timeline by their last notes, as shown in Fig. 3.6. The 3 hyperparameters are:

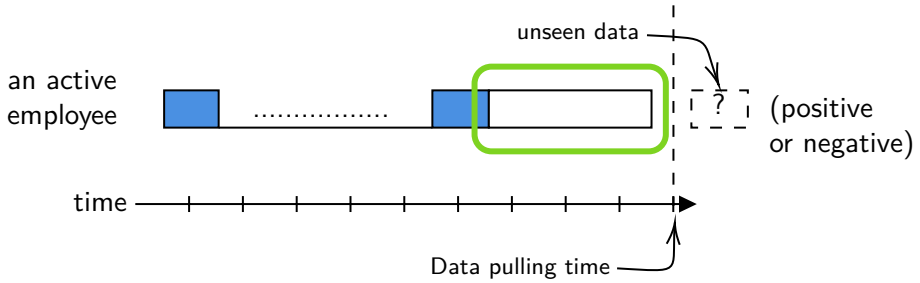


Fig. 3.4. Uncertainty for the unseen data of an active employee. For an active employee, there are always some notes which are written by the employee and close to data pulling time (within 90 days). Sometimes the notes quite approach to the data pulling time. This part of notes is indicated in the green box. After data pulling time, the employee's status is uncertain (active or terminated). The note marked by a question mark, after the data pulling time, can be either negative or positive. Since we assume there would be a transition phase between the negative and positive notes from a terminated employee, the labels of this part of notes can be uncertain.

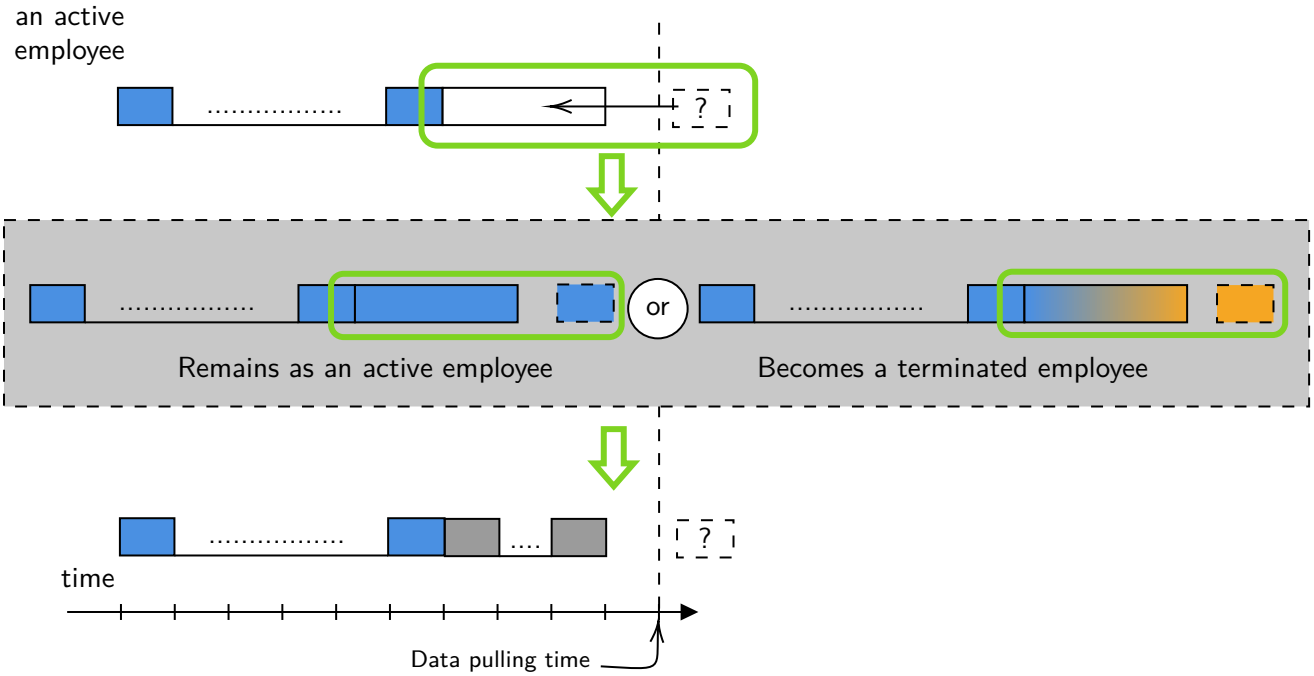


Fig. 3.5. Safety period for an active employee.

- n : The total number of weeks of data for each employee we bring into our dataset.
- p : The number of weeks of data labelled as positive data.
- st : The number of weeks of data in the safety phase.

We use a short code format to mention the hyperparameters easily. We illustrate it in Fig.3.7. The first digit is n in black; the second one is p in orange; and the third is st in grey.

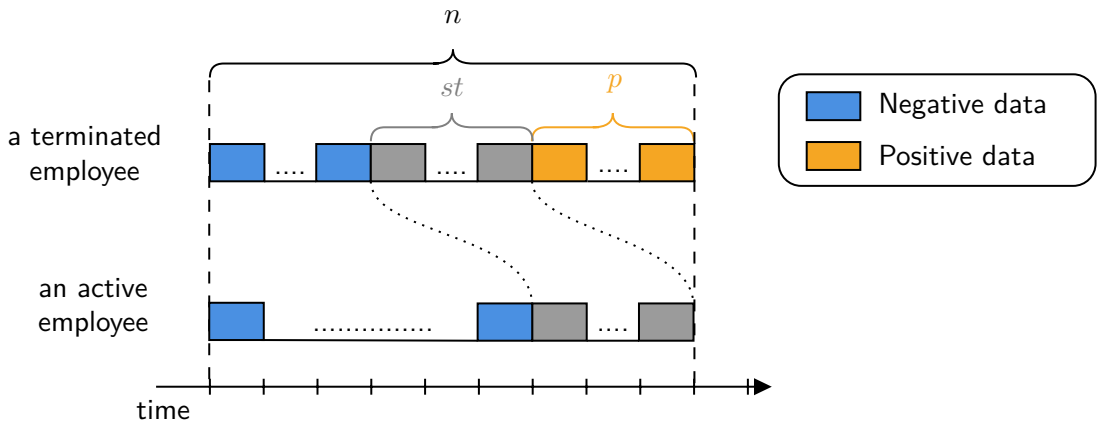


Fig. 3.6. The hyperparameters related to labelling and building the data set.

The value ranges of the hyperparameters are:

- $n \in \{1,2,3,4\}$
- $p \in \{1,2\}$
- $st \in \{1,2\}$

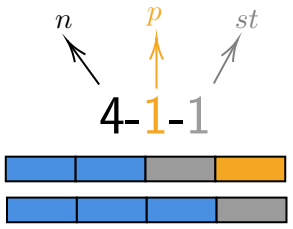


Fig. 3.7. An example of hyperparameters short code and simplified illustration

3.1.3. Tasks

Based on the labelings aforementioned, we designed 3 tasks.

The first one is to predict the class of each note. Formally, our data set is a sample S of m items:

$$S = \{(\boldsymbol{x}^{(i)}, y^{(i)}) | i = 1,...,m\} \tag{3.1.1}$$

where

- $\boldsymbol{x}$ is a **note** record represented as a vector in $\mathbb{R}^d$,
- y is the class and $y \in \{0,1\}$,
- m is the number of **notes**.

The notes are labelled by the method disclosed in Section 3.1.2. The task is that given a test **note** $\boldsymbol{x}$, we try to predict its true class y by:

$$\hat{y} = h(\boldsymbol{x}) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \{Pr(y|\boldsymbol{x})\} \tag{3.1.2}$$

where $h(\boldsymbol{x})$ is a linear binary classifier[1]. To identify this task more easily, we refer to it as: $Pred(Class|note)$.

Since notes are relatively short, in terms of both text length and time span, we group the notes of each employee by week. A weekly note group of one employee is called a **period** of notes. The second task is to predict the class of each period. We proposed this task based on these considerations:

- There's a significant correlation between the job dissatisfaction and weekly working hours. [3]
- The model will be triggered and executed weekly in AlayaCare system.
- The employees' shifts are normally scheduled weekly.

Similarly we have the dataset S as in Equation 3.1.1, but the differences are:

- $\boldsymbol{x}$ is a **period** of notes represented as a vector in $\mathbb{R}^d$,
- y is the class and $y \in \{0,1\}$,
- m is the number of **periods**.

We refer to this task as: $Pred(Class|period)$.

Lastly, we will predict the class of each employee. As a result, the dataset S as in Equation 3.1.1, but the differences are:

- $\boldsymbol{x}$ is a set of **employee**'s notes represented as a vector in $\mathbb{R}^d$,
- y is the class and $y \in \{0,1\}$,
- m is the number of **employees**.

We refer to this task as: $Pred(Class|employee)$.

3.2. Evaluation

3.2.1. Metrics

The receiver operating characteristic curve, or ROC curve is a plot with the true positive rate (TPR) against the false positive rate (FPR) when threshold is varied. Where

$$TPR\ (True\ Positive\ Rate)/Recall/Sensitivity = \frac{TP}{TP + FN}; \tag{3.2.1}$$

$$FPR = 1 - Specificity = 1 - \frac{TN}{TN + FP} = \frac{FP}{TN + FP}. \tag{3.2.2}$$

As shown in Fig. 3.8, the TPR is plotted on y axis and FPR is plotted on x axis. The y = x line (FPR = TPR) shows the curve corresponds to the performance of the random guessing. A single scalar value is proposed to represent the classifier performance since the ROC curve is a two-dimensional presentation. The area under the ROC curve (AUC) is a

[1]The reason that we choose linear classifier in this project is that, beside reporting the evaluation result for each task, we will analyze as well the feature importance by checking the feature weights of the classifier.

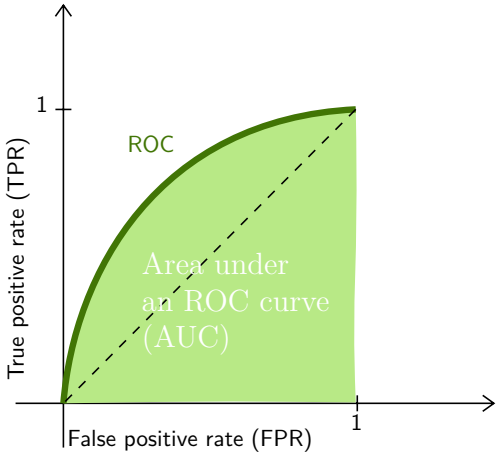


Fig. 3.8. An ROC curve and area under the curve.

commonly accepted method since Bradley [4]. The AUC is always in the range of [0,1.0]. The y = x line which corresponds to the random guessing will has an AUC of 0.5. The realistic classifiers should have an AUC larger than 0.5. Researchers often use AUC as a general measure of predictiveness and it works very well in general [10].

3.2.2. Cross-validation

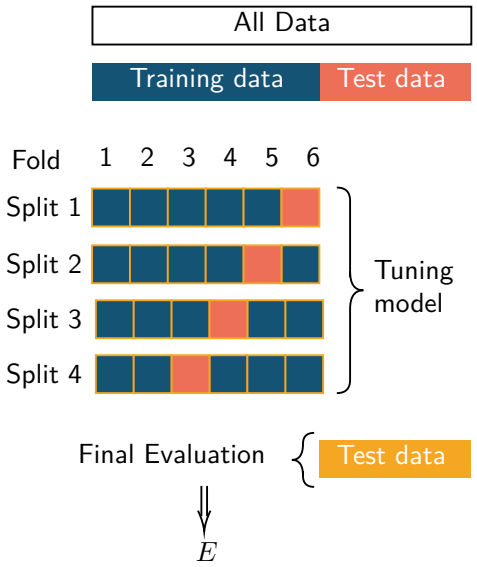


Fig. 3.9. Cross-validation illustration.

The labelling related hyperparameters not only plays a role in the optimization of our model, but also provides some useful information in the task of job analysis. During the early stage of the experiments, I found the model is quite easy to overfit for task $Pred(Class|period)$ and task $Pred(Class|employee)$ due to the small size of datasets.

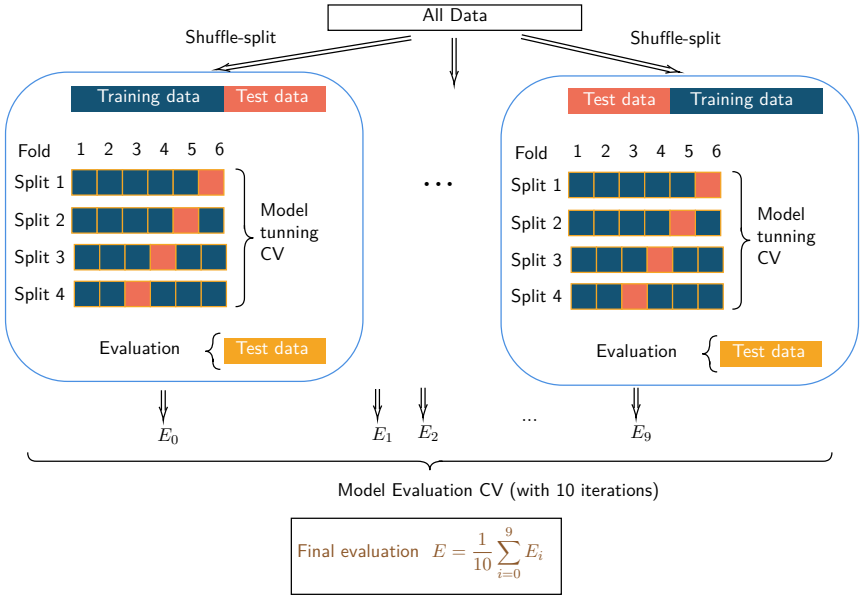


Fig. 3.10. Nested Cross-validation illustration

Nested cross-validation (or "double-cross") usually helps better optimize and evaluate the model as Stone [39] mentioned. Nevertheless, Krstajic [23] pointed out that there are even pitfalls in nested cross-validation, and suggested repeated v-fold (or k-fold) cross-validation, which is substantially more computationally intensive.

Cross-validation is one popular method to avoid overfitting. The common cross-validation is presented in Fig. 3.9. The nested cross-validation (CV) add another model evaluation CV outside of model tuning CV as shown in Fig. 3.10. During each iteration of model evaluation CV, we do: 1) Random split the data in to train and test; 2) Apply the model tuning CV to train and tune our model; 3) Evaluate on the unseen test data to get the evaluation E_i with the tuned model from 2). Eventually, when all the iterations of model evaluation CV terminates, we average all the E_i and get the final evaluation E.

In our experiments, the evaluation results from shuffle-split (random split) cross-validation converge eventually while the number of iterations increases after being greater than 10. For this reason, it is our choice through the whole project. Shuffle-split cross-validation will not generalize better than repeated cross-validation but it offers a choice between the v-fold and repeated v-fold cross-validation. It helps us avoid the small sub-sample problem in v-fold cross-validation when v is large and the infeasible computation in repeated v-fold cross-validation when the number of repetitions is large.

Fig. 3.11 shows the flow chart of the nested cross-validation used in this research.

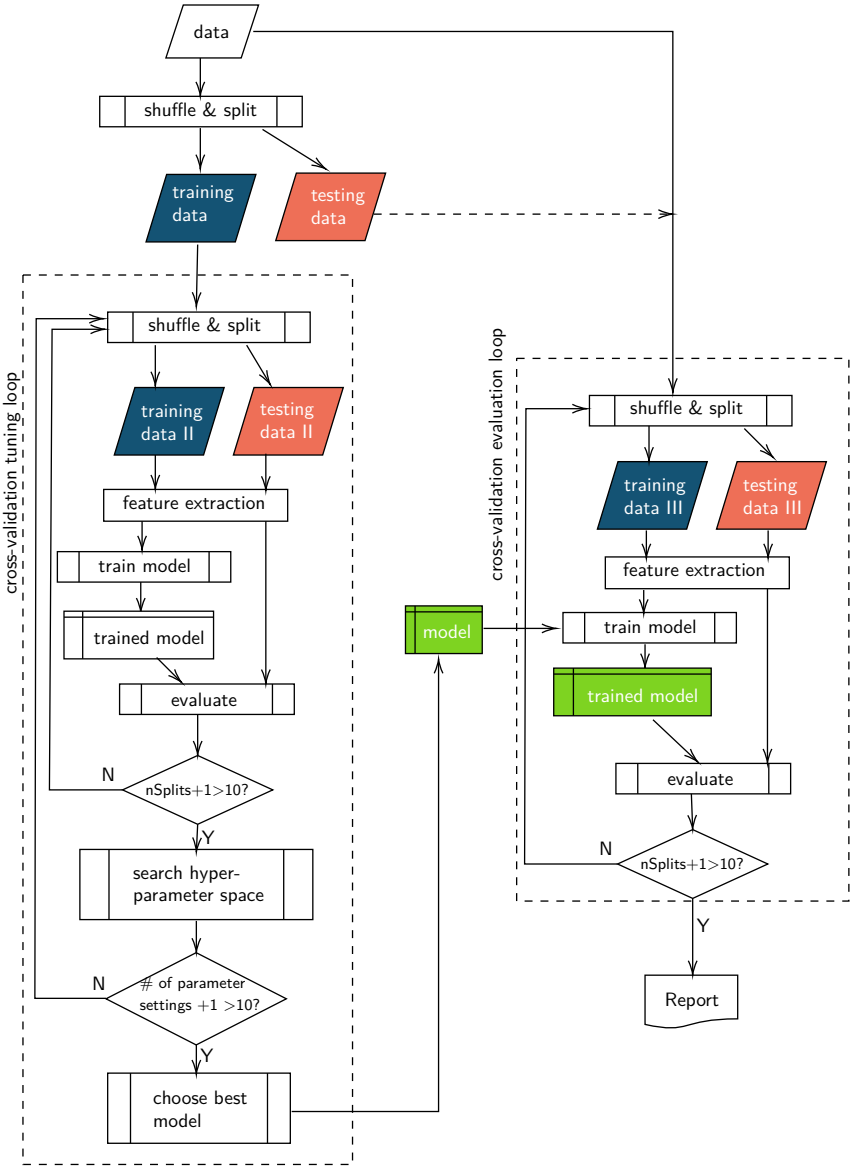


Fig. 3.11. The modelling process. (*nSplits* is the number of re-shuffling & splitting iterations.)

3.2.3. Dataset splits

In order to select train/test splits, we applied two kinds of shuffle-split. The strategies illustrated in Fig. 3.12. The grouped shuffle-split will make sure the data from same group won't show up in the training set and test set at same time.

3.2.4. Training and testing set preparation

There's a potential risk during stratified splitting. The models may be trained by the future data and tested on the ancient data. This data leakage will probably bring the future

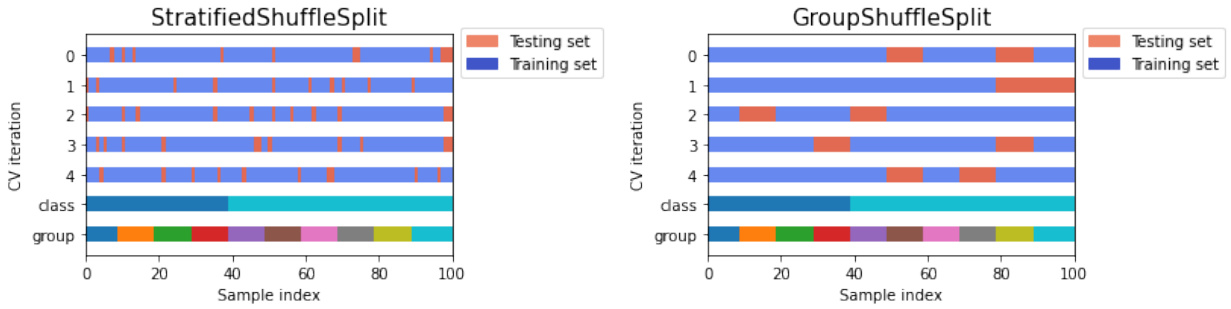


Fig. 3.12. Stratified shuffle-split and grouped shuffle-split. (Figure source: `https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html`)

information to the past in a classification task. A simple way to avoid this is that we split the data using grouped shuffle-splits, where the employees are the groups. Thus, the notes from the employees in the testing data won't show up in the training data and vice versa. Since the cross-validation is used to report the performance of our models, the train/test splitting is accomplished during the process of each iteration of the cross-validation. The model performance based on both stratified shuffle-split and grouped shuffle-split will be reported.

Task	Training:Testing	Dataset A		Dataset B	
		train	test	train	test
Pred(Class|Note)	80%:20%	72033	18008	30057	7514
Pred(Class|Period)	90%:10%	7640	849	6048	672
Pred(Class|Employee)	80%:20%	418	105	401	100

Table 3.1. Available samples of training and testing set for 3 tasks over two datasets.

Table 3.1 shows the available sample numbers for the 3 tasks in theory. For each specific experiment, the training and testing set size varies due to the hyper-parameter of each experiment. It should be noted that sample numbers are relatively small in task $Pred(Class|period)$ and task $Pred(Class|employee)$ because the samples are actually some bundles of concatenated notes.

In task $Pred(Class|note)$, 7 different n-p-st combinations will be used on both datasets. We took the last 8 weeks ($n = 8$) of notes of each user. It's around 30% of total dataset received. Table 3.2 listed the sample numbers for each experiment over both datasets.

n-p-st	Type	Dataset A		Dataset B	
		y=0	y=1	y=0	y=1
(8,1,1)	train+valid	26392	1554	10483	638
	test	6599	388	2622	159
(8,2,1)	train+valid	24543	3725	9645	1490
	test	6136	932	2411	373
(8,3,1)	train+valid	22588	5575	8884	2330
	test	5647	1394	2222	582
(8,4,1)	train+valid	20410	7530	8068	3089
	test	5102	1883	2017	773
(8,5,1)	train+valid	18382	9709	7255	3906
	test	4596	2427	1814	977
(8,6,1)	train+valid	16670	11736	6421	4719
	test	4168	2934	1605	1180
(8,7,1)	train+valid	15092	13448	5681	5554
	test	3773	3362	1421	1388

Table 3.2. Sample numbers for Task $Pred(\ Class\ |note\ )$

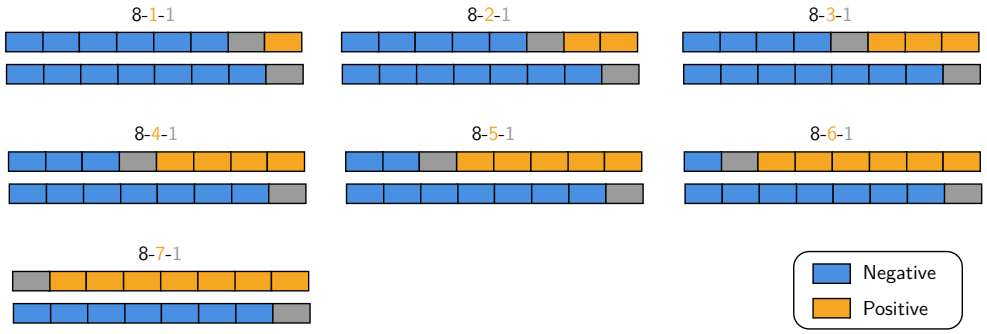


Fig. 3.13. The different combination of *n-p-st* with the short code for task $Pred(Class|note)$. E.g.: , 8-7-1 means $n = 8$, $p = 7$, $st = 1$.

Fig. 3.13, illustrates the *n-p-st* combinations used in this task with the short code format defined in Section 3.1.2. Fig. 3.14 shows positive and negative counts of training samples on different *n-p-st* combinations on two datasets. It shows, by decreasing the p value, the data become more and more imbalanced.

In task $Pred(Class|period)$, 11 different *n-p-st* combinations will be used on both Dataset A and B. Fig. 3.15, illustrates the *n-p-st* combinations used in this task. Table 3.3 listed the sample numbers for each experiment over both datasets. Fig. 3.16 and Fig. 3.17 show the train and test dataset counts (left) and the positive/negative percentage (right).

Comparing Fig. 3.16 to Fig. 3.17, it shows that these combinations of *n-p-st* lead generally to more well balanced datasets but with the same pattern on Dataset B. In both Fig. 3.16 and Fig. 3.17, the counts are the number of period (week) other than the number of notes,

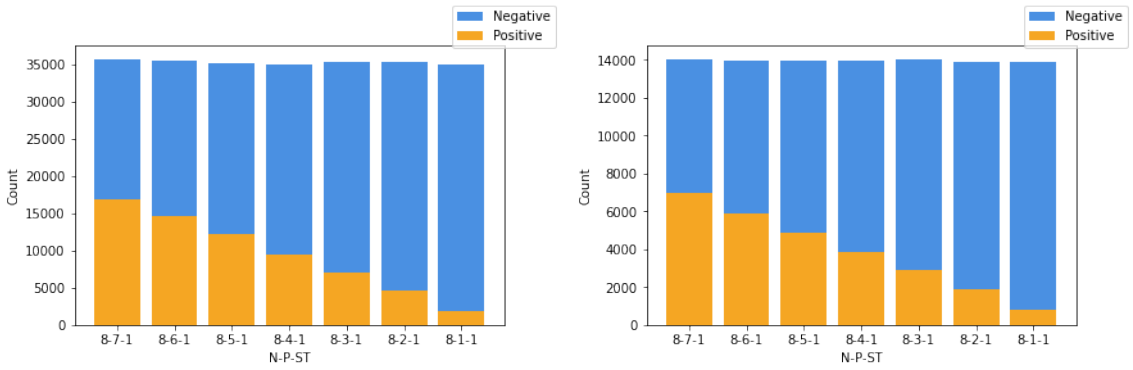


Fig. 3.14. Dataset A (left) and B (right): The visualization of the positive and negative counts of samples on different *n-p-st* combinations for task $Pred(Class|note)$.

n-p-st	Type	Dataset A		Dataset B	
		y=0	y=1	y=0	y=1
(8,7,1)	train+valid	941	1350	1010	1039
	test	235	338	210	305
(7,6,1)	train+valid	810	1231	867	939
	test	210	285	213	237
(6,5,1)	train+valid	698	1048	733	817
	test	175	262	175	219
(6,4,2)	train+valid	544	866	585	661
	test	151	200	140	177
(5,4,1)	train+valid	566	850	588	661
	test	136	216	140	177
(4,3,1)	train+valid	426	647	443	501
	test	102	166	105	134
(4,1,1)	train+valid	636	222	606	172
	test	154	58	149	46
(4,2,2)	train+valid	274	445	295	338
	test	76	106	70	90
(3,2,1)	train+valid	286	437	296	338
	test	68	114	70	90
(3,1,2)	train+valid	138	226	148	172
	test	38	54	35	46
(2,1,1)	train+valid	144	222	148	172
	test	34	58	35	46

Table 3.3. Sample numbers for Task $Pred(\ Class\ |period\ )$

since this task is to predict the class of instead of the class of notes. The number of notes covered in the 8-7-1 split is around 33k for Dataset A and 13k for Dataset B.

In task $Pred(Class|employee)$, 3 different MID-FIN combinations is tested on both datasets. Table 3.4 listed the sample numbers for each experiment over both datasets. As we increase the MID and FIN size, which means we observe a long historic and recent

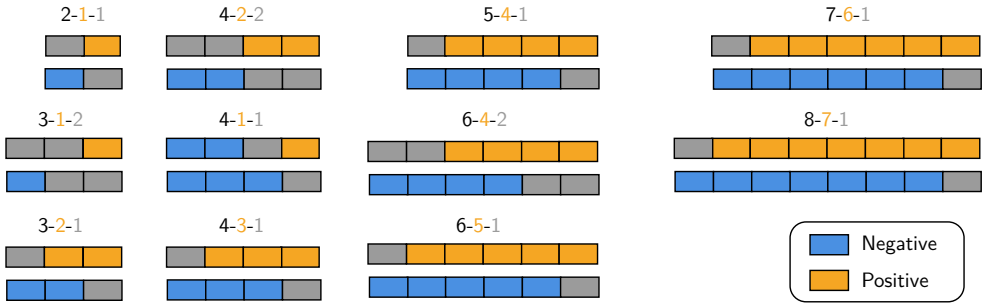


Fig. 3.15. The different combination of *n-p-st* with the short code for task Pred(Class|period). E.g.: , 8-7-1 means $n = 8$, $p = 7$, $st = 1$.

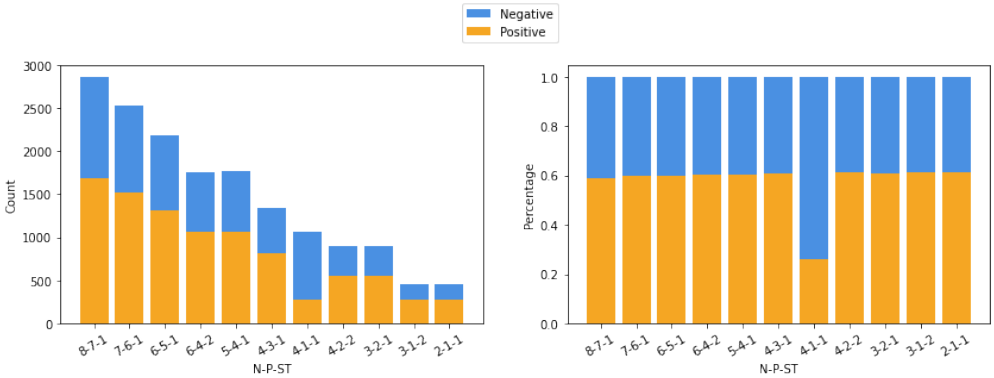


Fig. 3.16. Dataset A: The train and test sample counts (left) and percentage (right) on different *n-p-st* combinations for task $Pred(Class|period)$.

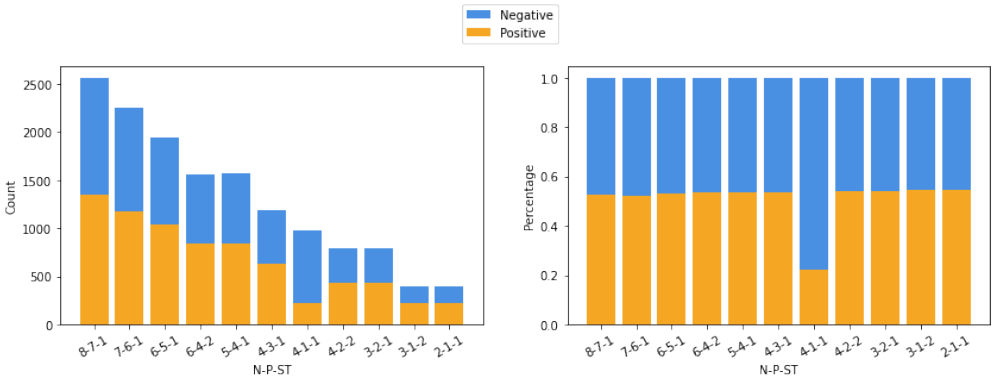


Fig. 3.17. Dataset B: The train and test sample counts (left) and percentage (right) on different *n-p-st* combinations for task $Pred(Class|period)$.

period of note history of a user, there would be some samples which are not qualified and be filtered out. Fig. 3.18 and Fig. 3.19 show the train and test dataset counts (left) and the positive/negative percentage (right). In all the cases, the data are quite balanced.

MID-FIN	Type	Dataset A		Dataset B	
		y=0	y=1	y=0	y=1
2-2	train+valid	139	219	147	176
	test	35	55	37	44
3-3	train+valid	137	195	144	158
	test	34	49	36	40
4-4	train+valid	130	170	136	143
	test	32	43	34	36

Table 3.4. Sample numbers for task $Pred(Class|employee)$

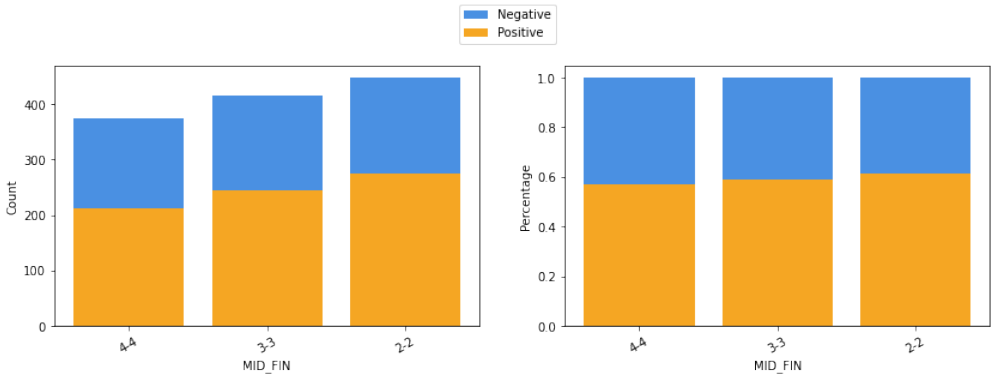


Fig. 3.18. Dataset A: The train and test sample counts (left) and percentage (right) on different $MID\text{-}FIN$ combinations for task $Pred(Class|employee)$.

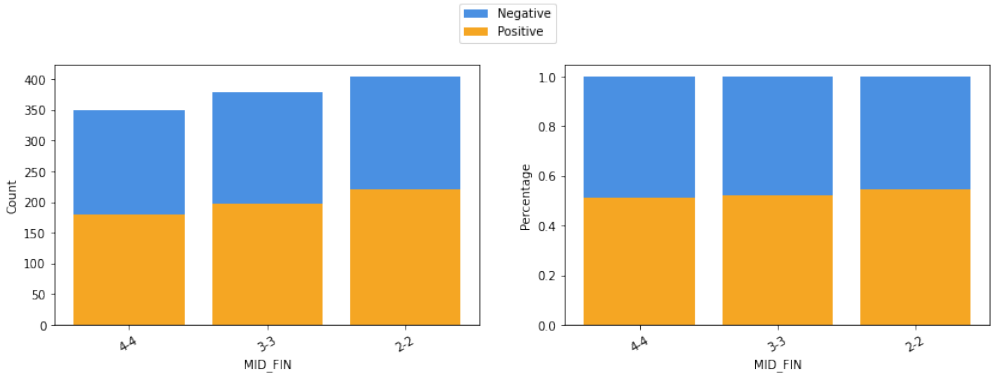


Fig. 3.19. Dataset B: The train and test sample counts (left) and percentage (right) on different $MID\text{-}FIN$ combinations for task $Pred(Class|employee)$.

Chapter 4

Models

4.1. Model paradigm

In this section, the structure and algorithm of each model will be reported. For task $Pred(Class|note)$, we designed a typical classification model. Fig 4.1 illustrates its structure while Algorithm 1 demonstrates it with a pseudo code.

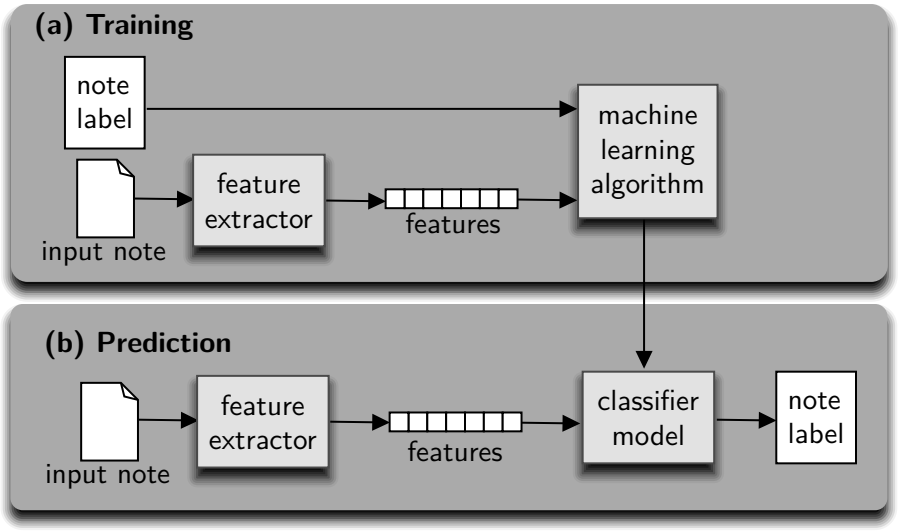


Fig. 4.1. The structure of the model for task $Pred(Class|note)$.

- During training, a feature extractor turns each input note to a feature set $\boldsymbol{x}$. ($\boldsymbol{x}$,y) are fed into the machine learning algorithm to generate a linear binary classifier $h(\boldsymbol{x})$, as mentioned in Equation 3.1.2.
- During prediction, the same feature extractor is used to turn unseen input note to feature sets. These feature sets are then fed into $h(\boldsymbol{x})$, which generates predicted labels.

Algorithm 1: The model for task $Pred(Class|note)$

Data: dataTrain, dataTest
Result: model_h

```
def TrainingProcess(dataTrain):                                  // (a) Training
    i ⟵ 0;
    foreach (note, label) in dataTrain do
        (X^(i), y^(i)) ⟵ (ExtractFeature(note), label);
        i ⟵ i + 1;
    end
    model_h.Fit(X, y);
    return model_h ;
;
def PredictionProcess(dataTest, model_h):                        // (b) Prediction
    i ⟵ 0;
    foreach note in dataTest do
        x ⟵ ExtractFeature(note);
        ŷ^(i) ⟵ model_h.Predict(x);
        i ⟵ i + 1;
    end
    return ŷ ;
```

For task $Pred(Class|period)$, we designed two models. The first one, model A, is a typical one. All the note text of one period are concatenated to form the input which will pass through the feature extractor. The structure of model is shown in Fig 4.2 and pseudo code is shown in Algorithm 2.

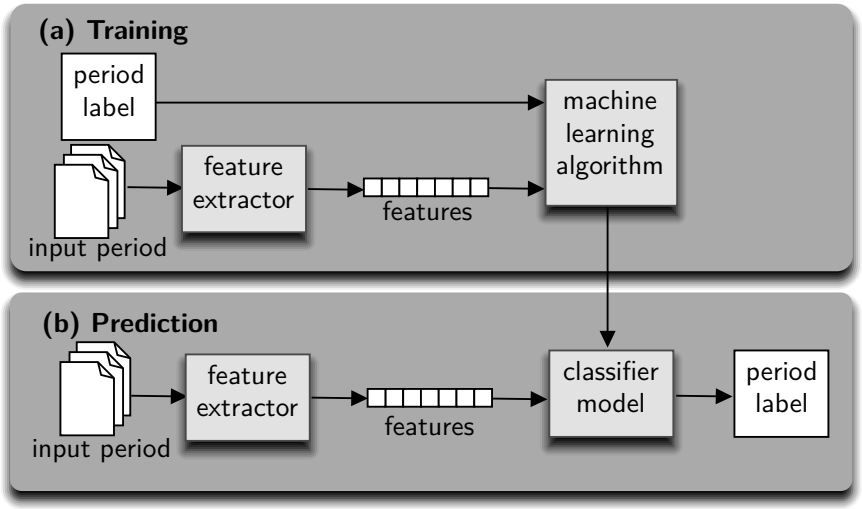


Fig. 4.2. The structure of model A for task $Pred(Class|period)$

Algorithm 2: Model A for task $Pred(Class|period)$

Data: dataTrain, dataTest
Result: model_h

```
def TrainingProcess(dataTrain):                          // (a) Training
    i ⟵ 0;
    foreach (period, label) in dataTrain do
        (X^(i), y^(i)) ⟵ (ExtractFeature(period), label);
        i ⟵ i + 1;
    end
    model_h.Fit(X, y);
    return model_h ;
;
def PredictionProcess(dataTest, model_h):                // (b) Prediction
    i ⟵ 0;
    foreach period in dataTest do
        x ⟵ ExtractFeature(period);
        ŷ^(i) ⟵ model_h.Predict(x);
        i ⟵ i + 1;
    end
    return ŷ ;
```

The second model for task $Pred(Class|period)$, model B is a variant of model A. The structure of model is shown in Fig 4.3 and pseudo code is shown in Algorithm 3.

- During training, a feature extractor turns each input note to a feature set $\boldsymbol{x}$. ($\boldsymbol{x}$,y) are fed into the machine learning algorithm to generate a linear binary classifier $h(\boldsymbol{x})$.
- During prediction, each note included in the unseen period will pass through the same feature extractor, and be turned into a group of feature sets. These feature sets are then fed into $h(\boldsymbol{x})$, which generates predicted labels for each note. Subsequently, all the note labels within that period will do the "majority vote" and output the predicted period label.

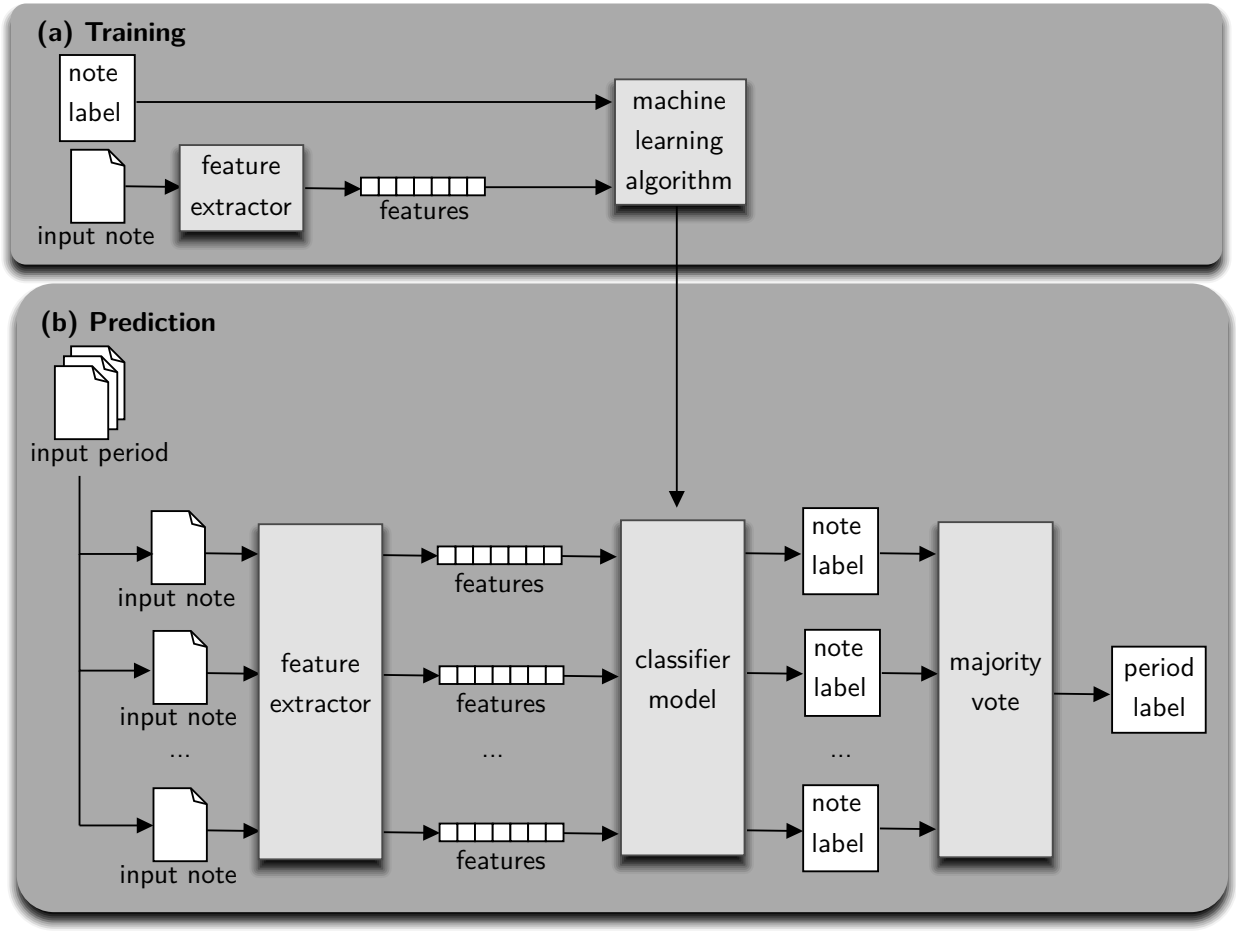


Fig. 4.3. The structure of model B for task $Pred(Class|period)$

Algorithm 3: Model B for task $Pred(Class|period)$

Data: dataTrain, dataTest

Result: model_h

```
def TrainingProcess(dataTrain):                                  // (a) Training
    i ⟵ 0;
    foreach (period,  label) in dataTrain do
        foreach note in period do
            (X^(i),  y^(i)) ⟵ (ExtractFeature(note),  label);
            i ⟵ i + 1;
        end
    end
    model_h.Fit(X, y);
    return model_h ;
;
def PredictionProcess(dataTest, model_h):                        // (b) Prediction
    i ⟵ 0;
    foreach period in dataTest do
        (votes,  j) ⟵ (0,  0);
        foreach note in period do
            x ⟵ ExtractFeature(note);
            ŷ ⟵ model_h.Predict(x);
            votes ⟵ votes + ŷ;
            j ⟵ j + 1;
        end
        ŷ^(i) ⟵ round(votes/j);                                  // majority vote
        i ⟵ i + 1;
    end
    return ŷ ;
```

For task $Pred(Class|employee)$, we use 2 types of periods: MID period and FIN period. MID periods located in the middle and FIN periods at the end of an employee's note time span. MID and FIN are two integer hyperparameters. The MID[1] periods contribute some contextual attributes to the features meanwhile the FIN periods bring in some triggering attributes. All the note text of one type of period are concatenated to form the input which will pass through the feature extractor. The features extracted from these two kinds of period will be eventually concatenated. For one sample, suppose $MID\ feature = (x_1,x_2,...,x_d)$, $FIN\ feature = (x'_1, x'_2,...,x'_d)$, then the final feature is:

$$Final\ feature = (x_1,x_2,...,x_d,x'_1, x'_2,...,x'_d)$$

Fig. 4.4 illustrates these two kinds of periods. This model is illustrated in Fig.4.5.

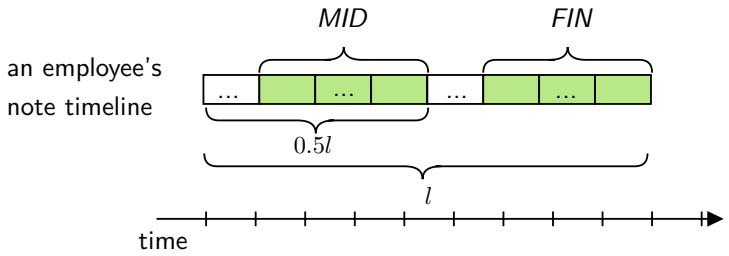


Fig. 4.4. The MID and FIN periods in the model for task $Pred(Class|employee)$. By putting an employee's all the notes on a timeline, we mark the time span of the notes as l which indicates the time duration between the created time of employee's first note and the last note in the dataset. $0.5l$ indicates the time duration between the created time of employee's first note and the "middle point" note, which is located or close to the middle point of the time span. We call the periods which include last note and the "middle point" note as last period and middle period respectively. Suppose this employee's note include q periods in total. The periods are sequentially numbered as 0,1,2,...,p,...,q where p is the id of middle period and q is the id of last period. The periods whose id are from $\{p - MID - 1,\ p - MID - 2,\ \ldots,\ p - 1,\ p\}$ are marked as MID periods. The periods whose id are from $\{q - FIN - 1,\ q - FIN - 2,\ \ldots,\ q - 1,\ q\}$ are marked as FIN periods.

[1]We use MID and FIN to mention the value of the hyperparameter, while using MID and FIN to mention the type of the period.

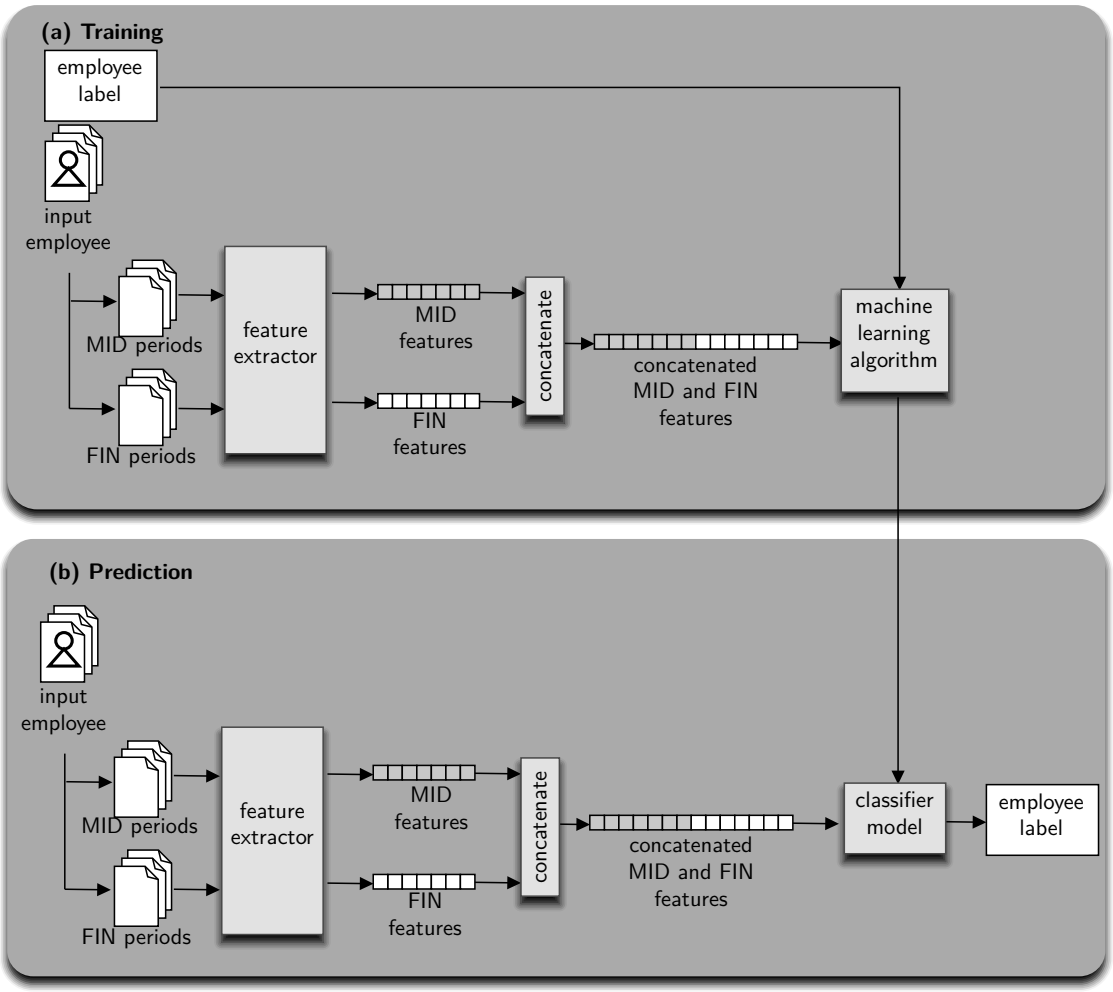


Fig. 4.5. The structure of the model for task $Pred(Class|employee)$

Algorithm 4: The model for task $Pred(Class|employee)$

Data: dataTrain, dataTest

Result: model_h

```
def TrainingProcess(dataTrain):                                  // (a) Training
    i ⟵ 0;
    foreach (employee,  label) in dataTrain do
        x_MID ⟵ ExtractFeature(employee.MID_periods);
        x_FIN ⟵ ExtractFeature(employee.FIN_periods);
        (X^(i),  y^(i)) ⟵ (Concatenate(x_MID, x_FIN),  label);
        i ⟵ i + 1;
    end
    model_h.Fit(X, y);
    return model_h ;
;
def PredictionProcess(dataTest, model_h):                        // (b) Prediction
    i ⟵ 0;
    foreach employee in dataTest do
        x_MID ⟵ ExtractFeature(employee.MID_periods);
        x_FIN ⟵ ExtractFeature(employee.FIN_periods);
        x ⟵ Concatenate(x_MID, x_FIN);
        ŷ^(i) ⟵ model_h.Predict(x);
        i ⟵ i + 1;
    end
    return ŷ ;
```

4.2. Features extraction

We considered a number of features typically used in related works.

4.2.1. Statistical Features

We selected 6 features in total: 5 basic statistical features mentioned by Kriz et al. [22] in the "Text length characteristics" part; 1 feature related to punctuation. They are:

- Number of sentences
- Number of tokens
- Number of characters
- Average sentence length (Number of tokens divided by number of sentences)
- Average token length (Number of characters divided by number of tokens)
- Number of quotation marks

Kriz et al. used the first 5 basic features in the task of Native Language Identification. It shows that these features are able to measure the different writing styles. We believe that when employees write the progress notes with dissatisfaction, their writing styles would also be different. The employees quote sometimes directly the patients' words. They implicitly show their agreement or disagreement of patient's words in this way. For this reason, we added the number of quotation marks as a feature as well.

4.2.2. VADER Features

VADER is a popular rule-based model for sentiment analysis tasks. It performs even better than individual human raters in one of its authors' experiments [19]. VADER brings us 4 features:

- Positive (*pos*) score
- Neutral (*neu*) score
- Negative (*neg*) score
- Compound (*compound*) score

The *pos*, *neu*, and *neg* scores are percentages for parts of text (tokens, expressions, punctuations) that belong to each category. And $pos + neu + neg = 1$. The *compound* score is computed by adding the adjusted sentiment ratings scores[2] of each word, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive), as shown in Fig 4.6.

Jung and Suh [21] showed the importance between the sentiment and 9 job satisfaction factors they spotted from online employee reviews. It shows that the sentiment analysis relates to the dissatisfaction detection.

[2]From 10 independent human raters.

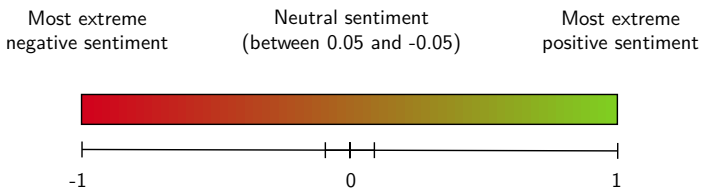


Fig. 4.6. The normalized VADER compound score

4.2.3. LIWC Features

Linguistic Inquiry and Word Count (LIWC) 2015 dictionary consists 73 categories. Furthermore it contains about 6600 words and stems. Each of them belong to one or more categories. Pennebaker claims that LIWC2015, on average, holds over 86% of the words people use [33].

There are about 260 words on average in each category. But as shown in Fig 4.7, the numbers varies a lot.

Fig. 4.7. The number of words/stems which belong to each category in LIWC2015

In this project, for each category, I count the words of a given text that fall into the category and divide the count by the total number of words of the text.

De Choudhury and Counts [6] showed that it is effective to measure positive affect and negative affect by LIWC. And the work-life imbalance dissatisfaction could cause the positive affect decreasing after-hours. This shows that LIWC may be a measurement of job dissatisfaction in some cases.

4.2.4. Language Model Features

Language model is a statistical model. Given a sentence $w_1,\ldots,w_m$, a language model gives a probability $P(w_1,\ldots,w_m)$ to this sentence. By applying the chain rule, we have:

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i \mid w_1,\ldots,w_{i-1})$$

With a n-gram approximation, it can be rewritten as $\prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)},\ldots,w_{i-1})$.

I take a short sentence from the notes as an example.

Example 1.

```
<s> Received client awake @21:40hr nurse given night meds .  </s>
 1      2       3     4       5      6     7    8      9  10  11
```

It's a sentence with 11 terms including the inserted `<s>` and `</s>`. They indicate respectively the start and the end of a sentence. That is to say, that we observe the preceding $n-1$ words instead of the preceding $i-1$ words when we calculate the probability of the i^{th} word. For example, as shown in Table 4.1, the probability of the term "given" is calculated as p(given | nurse) with bigram[3] other than p(given | <s> Received client awake ... nurse).

Position	Term	Log Probability	n-gram Context
1	<s>	NA	NA
2	received	-0.70	p (received | <s>)
3	client	-0.18	p (client | <s> received)
4	awake	-1.42	p (awake | <s> received client)
5	@21:40hr	-5.53	p (@21:40hr)
6	nurse	-2.81	p (nurse)
7	given	-2.43	p(given | nurse)
8	night	-1.10	p (night | nurse given)
9	meds	-0.61	p (meds | nurse given night)
10	.	-0.81	p (. | night meds)
11	</s>	-1.94	p (</s> | .)

Table 4.1. The probability of each term in Example 1. The colour of position number indicates the corresponding bin in Fig 4.8.

[3] n-gram, where n=2

After calculating the log probability of each term in a sample by the language model, I chose 8 features: 3 statistical metrics and 5 slice-score based on these probabilities. The 3 statistical metrics are: average, max and min of the term probabilities. The 5 slice-score ($SLC_1, \ldots, SLC_5$) is simply the number of terms that fall into a specific bin of the histogram of term probabilities. The histogram of Example 1 is shown in Fig 4.8 while the terms belonging to each bin are shown in Table 4.2. Table 4.3 shows the 8 features of Example 1.

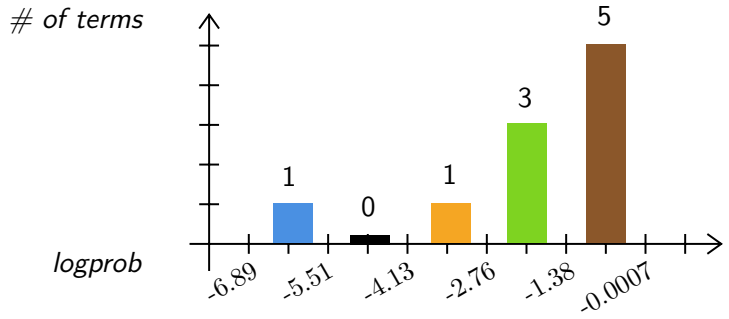


Fig. 4.8. The histogram of term probability in Example 1. (Corpus max = -0.0007, corpus min = -6.89)

Bin Number	Bin Boundary	Term Position
#1	[-6.89, -5.51]	5
#2	[-5.51, -4.13]	-
#3	[-4.13, -2.76]	6
#4	[-2.76, -1.38]	4, 7, 11
#5	[-1.38, -0.0007]	2, 3, 8, 9, 10

Table 4.2. The details of the bins in histogram Fig 4.8.

Feat 1	Feat 2	Feat 3	Feat 4	Feat 5	Feat 6	Feat 7	Feat 8
average	max	min	SLC_1	SLC_2	SLC_3	SLC_4	SLC_5
-1.75	-0.18	-5.53	1	0	1	3	5

Table 4.3. The 8 features for Example 1 without normalization.

The histogram is able to summarize the information and the language model is able to provide the likelihood of the terms. Kriz et al. [22] used the language model with a cross-entropy score in the task. We believe this feature set will help us to detect the dissatisfaction as well.

4.2.5. TF-IDF Features

TF-IDF (Term frequency and Inverse document frequency) is a statistic that shows the importance of a word to a document in a corpus. Briefly, TF-IDF is the product of TF and IDF. It highlights the terms that appear often in a document but not so in the corpus.

Without limitation, the dimension of TF-IDF features is the length of the vocabulary of the corpus. It's usually in the thousands.

In this project, I choose the top 500 terms ordered descendingly by their frequency across the corpus since the number of terms lager than 500 doesn't bring a significant performance increase.

TF-IDF is widely used in the classifications of the satisfaction/dissatisfaction text, e.g., Forster and Entrup [13], Rehan et al [35].

4.2.6. DistilBERT Features

Bidirectional Encoder Representations from Transformers (BERT) [7] is a pretrained NLP model developed by Google in 2018. It relies on a stack of transformers. DistilBERT [36] is a model pretrained by a simplified BERT base. The authors claimed it's 40% smaller than BERT and 60% faster at inference time while keeping the almost the same performance of BERT [36].

The vectorized output with a dimension of 768 from the DistilBERT model is taken directly as the features.

BERT model performs well on NLP tasks which relates to the satisfaction/dissatisfaction [31] [12].

Chapter 5

Experiments

This chapter describes our experiments trying to resolve the 3 tasks mentioned in Section 3.1.3. The analysis will be presented in Chapter 6.

With Dataset A and B, the train-test sets are prepared separately for each task. The "stratified" and "group" indicate the train/test splitting method. As we mentioned in section 3.2.2, the "group" method makes sure the same user's notes in testing dataset won't show up in training dataset. They show obviously the "stratified" method brings better performance than the "group" method does. Generally, it will be harder for the model to tell the true class of a user's note without training by this user's notes. In other words, a user's notes in the training data will facilitate classifying the same user's notes in the testing data.

Suppose that we arbitrarily choose one note from the training data and one note from the testing data from the same user. We assume there are no notes that were written by the same user at the exact same time, then these two notes will be with two different timestamps. With the "stratified" method, there would be two situations for these two notes:

- Situation I: Past training note and future testing note[1]
- Situation II: Past testing note and future training note

Meanwhile, with "group" method, none of the two situations exist. From a perspective of a prediction system which is based on temporal data splits, the situation II should definitely be excluded. For this reason, we believe the score of "stratified" splits is greater than the score of temporal splits, which is greater than the score of "group" splits, by simply comparing the metrics of their classification performance. In other words, it shows the performances of the "stratified" and "group" splits are the upper and lower theoretical performance boundaries of performance of the temporal splits respectively.

In section 3.1.2, three variables n,p and st were defined. For different combination of n-p-st:

- Different number of samples will be brought into the training and testing.

[1]The note from the training data is written earlier than the note from the testing data.

- Different number of samples will be labelled as the positive or negative.

In addition to, the degree of the imbalance is dominated by the combinations of *n-p-st*.

- The small n value will lead to a small dataset.
- The small p value will lead to an imbalanced dataset split.
- The *st* value will lead to a "shift" of the negative data sampling.

By using this short code representation, it helps visualize these combinations in different tasks (Fig. 3.13 and Fig. 3.15). The maximum n in our experiments is 8, which represents 8 weeks due to the computational power is limited, larger n asks for more cpu and memory resources.

5.1. Task Pred(Class|note)

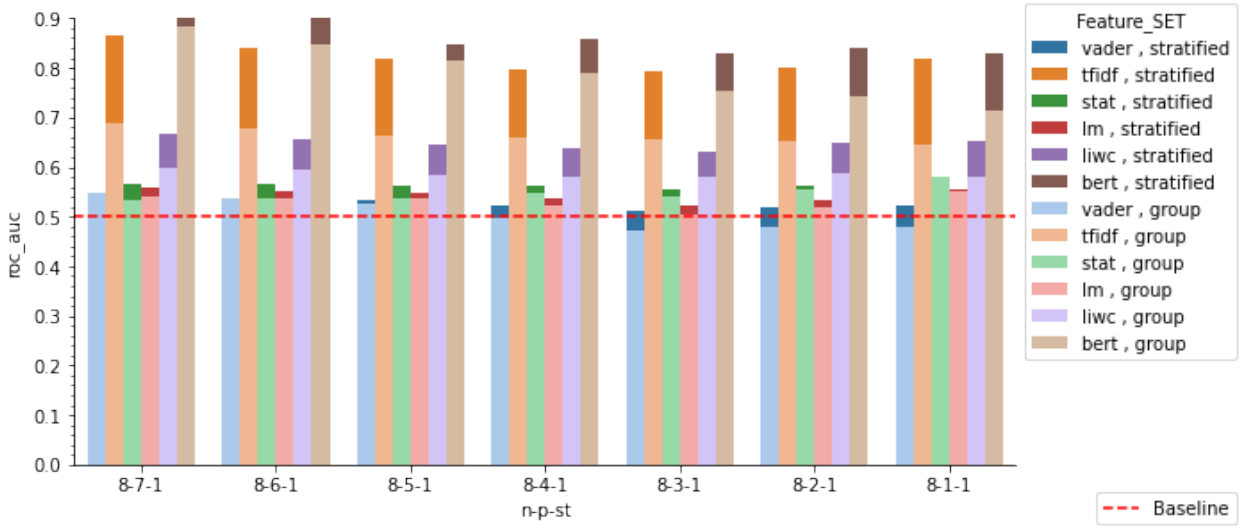


Fig. 5.1. Dataset A: AUC score with different splitting methods on different *n-p-st* combinations. For each *n-p-st* combination, 12 scores are reported (6 feature sets and 2 splitting methods). The AUC socre of the same feature set is indicated by the same color. The scores of the different splitting methods but same feature set are distinguished by the two different brightness of the color.

Fig. 5.1 presents the AUC score of this classification task on Dataset A. Generally the performance of stratified split method is better than that of grouped split. It shows also the "bert" feature outperforms other features for Dataset A. The performance of "tfidf" feature is not farm from that of "bert". The feature "liwc" is slightly better than the baseline ($AUC = 0.5$), which is the performance of the random guess. The left 3 features, which are "vader", "stat" and "lm" doesn't seem any better than the random guess. As we increase the p value, the difference between the performance of stratified "bert" and grouped "bert" shows a trend that it becomes smaller.

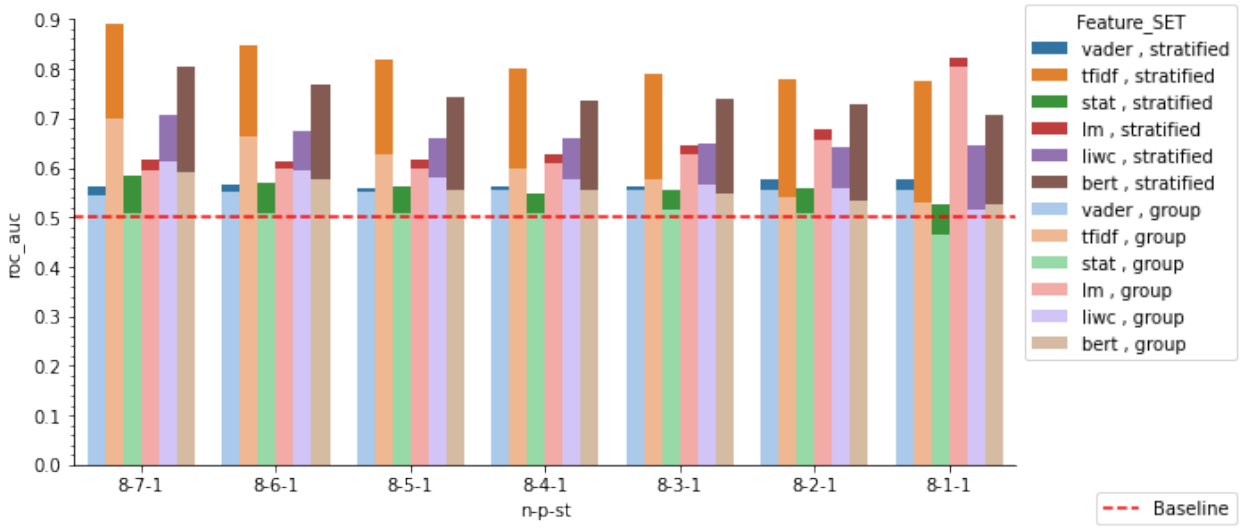


Fig. 5.2. Dataset B: AUC score with different splitting methods on different *n-p-st* combinations

Fig. 5.2 presents the AUC score of this classification task on Dataset B. It shows generally the "tfidf" feature is the best with stratified split method for Dataset B. With grouped split method, "tfidf" feature provides better performance over $5 \leq p \leq 7$ and "lm" feature provides better performance over $1 \leq p \leq 4$. Surprisingly, as we decrease the p value, the "lm" feature shows a trend that it provides a better performance.

n-p-st	Dataset A			Dataset B				
	Stratified		Grouped	Stratified				Grouped
	bert	tfidf	bert	bert	tfidf	liwc	lm	lm
8-1-1	.83♦	.82♦	.71◊	.71◊	.77◊	.65	.82♦	.80♦
8-2-1	.84♦	.80♦	.74◊	.73◊	.78◊	.64	.68	.65
8-3-1	.83♦	.79◊	.76◊	.74◊	.79◊	.65	.65	.63
8-4-1	.86♦	.79◊	.79◊	.74◊	.80♦	.66	.63	.61
8-5-1	.85♦	.82♦	.82♦	.74◊	.82♦	.66	.62	.60
8-6-1	.92★	.84♦	.85♦	.77◊	.85♦	.67	.61	.60
8-7-1	.90★	.86♦	.88♦	.80♦	.89♦	.71◊	.62	.60
◊'s count	0	2	4	**6**	3	1	0	0
♦'s count	5	**5**	**3**	1	**4**	0	1	1
★'s count	**2**	0	0	0	0	0	0	0
Ability	Outstanding	Excellent	Excellent	Acceptable	Excellent	-	-	-

Table 5.1. All the experiments with $AUC \geq 0.7$. (◊ : $0.7 \leq AUC < 0.8$, ♦ : $0.8 \leq AUC < 0.8$, ★ : $AUC > 0.9$.)

Hosmer's general guidelines [18] suggests:

- ($0.5 < AUC < 0.7$) is not better than a coin toss;
- ($0.7 \leq AUC < 0.8$) is acceptable discrimination;

- ($0.8 \leq AUC < 0.9$) is excellent discrimination.
- ($AUC \geq 0.9$) is outstanding discrimination.

According to this guidelines, we rank the feature sets' description ability and show them in Table 5.1. The rank we used are Outstanding ($\bigstar$: $AUC > 0.9$), Excellent ($\blacklozenge$: $0.8 \leq AUC < 0.8$) and Acceptable ($\lozenge$: $0.7 \leq AUC < 0.8$). The order is Outstanding > Excellent > Acceptable. One feature-split's ability will be ranked with the highest rank whose count greater than 1. For example:

- For Dataset A, bert-stratified provides ($\lozenge$'s count=0, $\blacklozenge$'s count=7, $\bigstar$'s count=2). The highest rank with the count greater than 1 is $\bigstar$ (Outstanding). So it is ranked as "Outstanding".
- For Dataset B, bert-stratified provides ($\lozenge$'s count=6, $\blacklozenge$'s count=1, $\bigstar$'s count=0). The highest rank with the count greater than 1 is $\lozenge$ (Acceptable). So it is ranked as "Acceptable".

For Dataset B, the bert-stratified, lm-stratified and lm-grouped only reports the $0.8 \leq AUC < 0.9$ once ($\blacklozenge$'s count is 1). Similarly, for Dataset B, the liwc-stratified only reports the $0.7 \leq AUC < 0.8$ once ($\lozenge$'s count is 1). It shows some randomness in these cases. For this reason, we don't mark the rank based on these $count = 1$ situations.

Feature-Split	Dataset A	Dataset B
bert-stratified	**Outstanding**	Acceptable
bert-group	**Excellent**	-
tfidf-stratified	**Excellent**	**Excellent**
tfidf-group	-	-

Table 5.2. Summarised feature-split ability of description for those provides at least acceptable description.

To clearly show the different feature-splits' ability of description, we take the ability values in Table 5.1 and list them separately in Table 5.2. All the ability values that are better than "acceptable" are on bold. It shows clearly that the models with different feature-splits work better on Dataset A.

5.2. Task Pred(Class|period)

In this task, for both datasets, the AUC score from 2 models with 6 feature sets by dozens of different *n-p-st* combinations will be shown in 4 separate figures. The experiments results on Dataset A are illustrated in Fig. 5.3 (model A) and Fig. 5.4 (model B). Respectively the experiments results on Dataset B are illustrated in Fig. 5.5 (model A) and Fig. 5.6 (model B). The ROC bars with the splitting method of "group" are stacked on the bars of the "stratified".

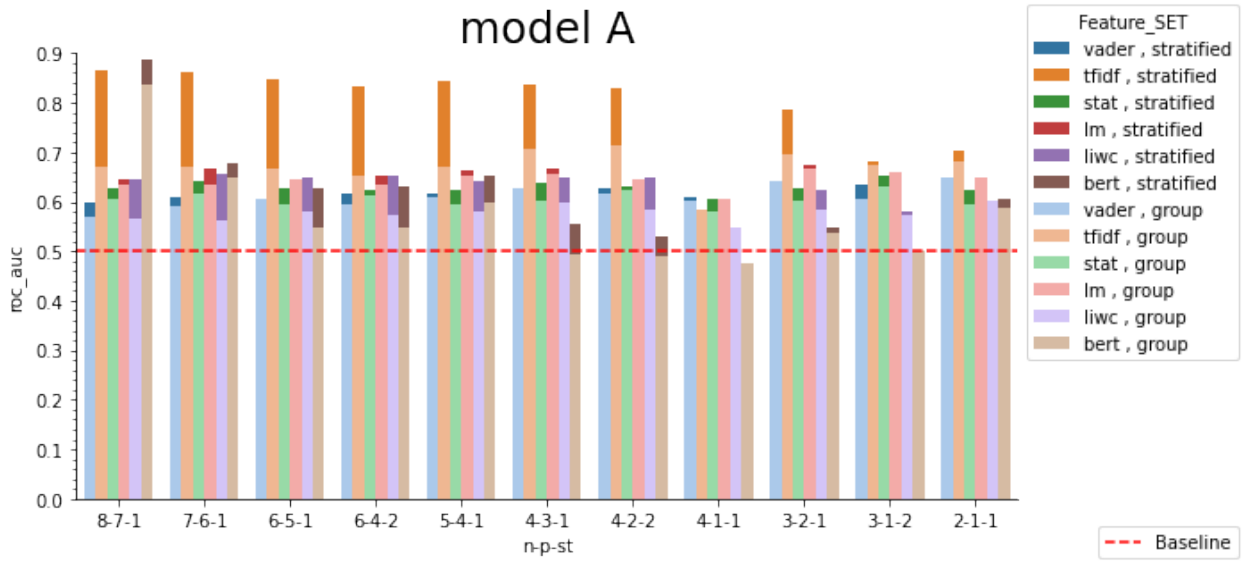


Fig. 5.3. Dataset A: AUC score by model A with different splitting methods on different *n-p-st* combinations

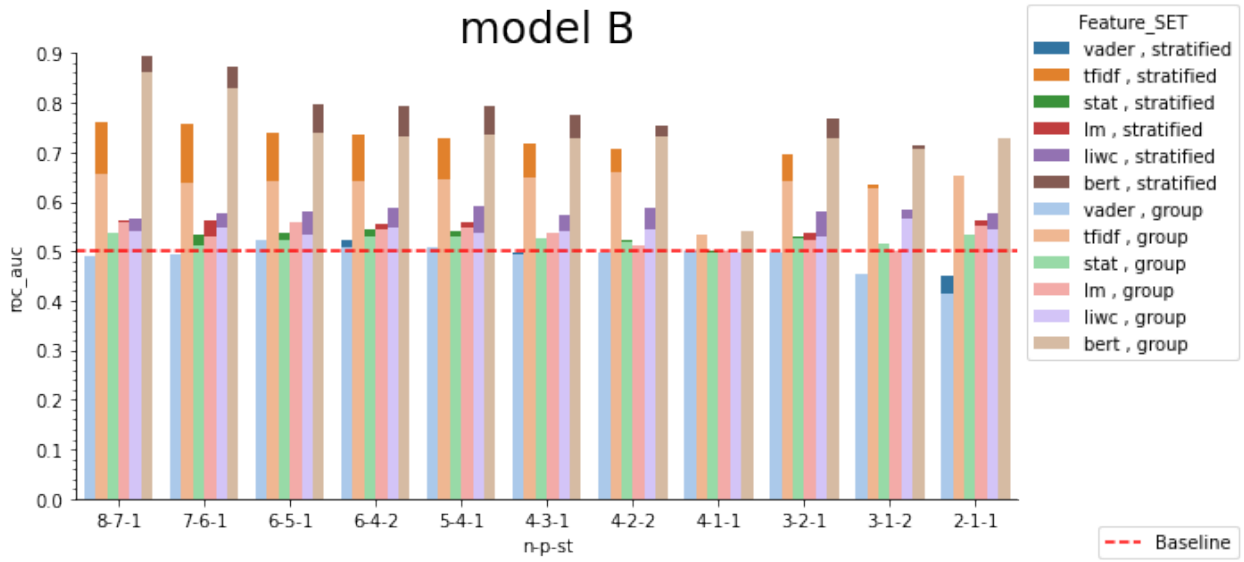


Fig. 5.4. Dataset A: AUC score by model B with different splitting methods on different *n-p-st* combinations

It shows the "stratified" method's results generally outperform the "group" method's. And the experiments with feature set "tfidf" generally outperforms other features sets on both datasets, except the model B on Dataset A. In most cases, the "group" splits gives a lower AUC score than "stratified" one.

In Fig. 5.3, Fig. 5.4, Fig. 5.5 and Fig. 5.6, we noticed:

- Generally bigger the dataset gives better performance.

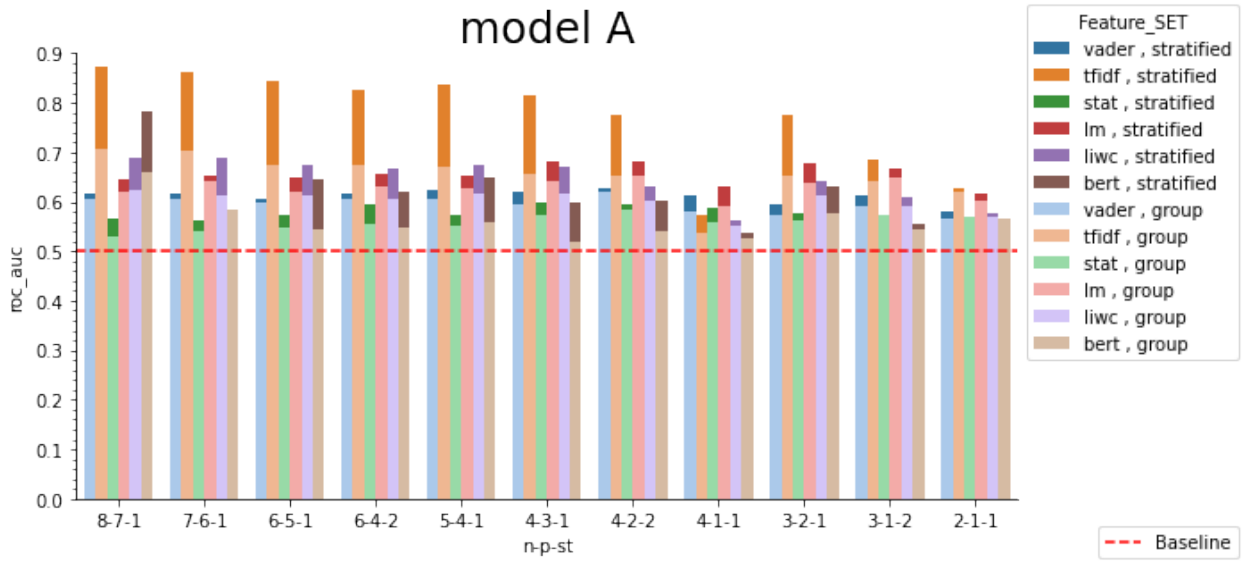


Fig. 5.5. Dataset B: AUC score by model A with different splitting methods on different *n-p-st* combinations

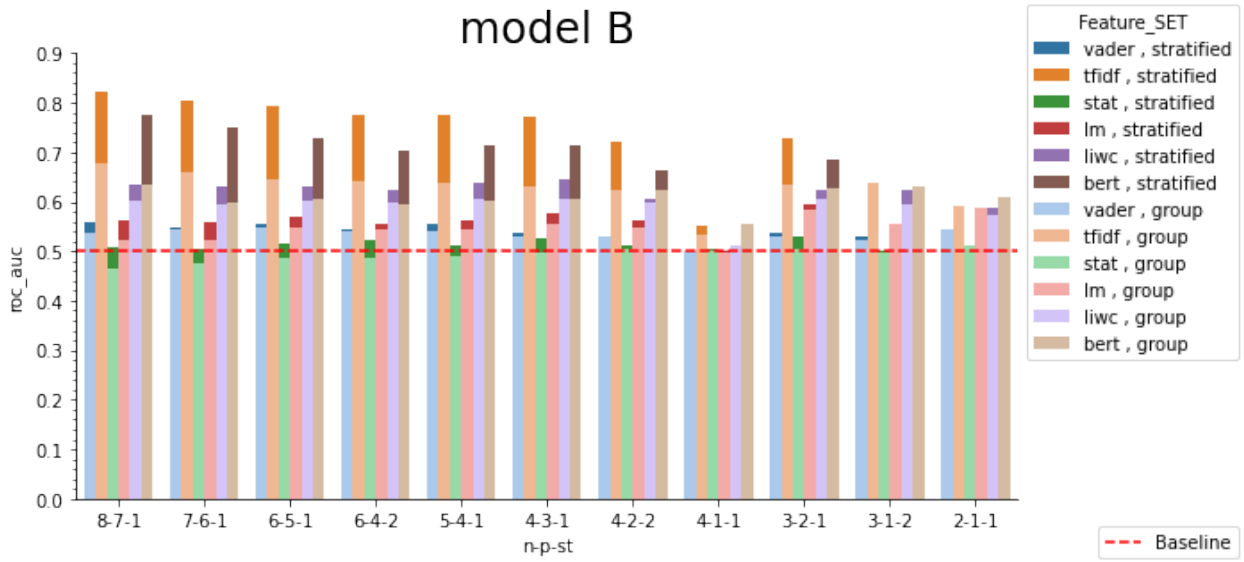


Fig. 5.6. Dataset B: AUC score by model B with different splitting methods on different *n-p-st* combinations

- The 4-1-1 presents a low AUC score, which indicates that the both model A and B don't work well with imbalanced splits.
- The *st* seems less important than *n* and *p*. In other words, "shift" the sampling of negative data doesn't affect performance. E.g.:
  - 2-1-1 and 3-1-2 yield the similar performance.
  - 3-2-1 and 4-2-2 yield the similar performance.
  - 5-4-1 and 6-4-2 yield the similar performance.

n-p-st	Dataset A				Dataset B			
	model A	model B			model A		model B	
	Stratified	Stratified		Grouped	Stratified	Grouped	Stratified	
	tfidf	bert	tfidf	bert	tfidf	tfidf	bert	tfidf
2-1-1	.70◊	.72◊	.64	.73◊	.63	.62	.60	.59
3-1-2	.68	.71◊	.63	.71◊	.68	.64	.61	.61
3-2-1	.78◊	.77◊	.69	.73◊	.77◊	.65	.68	.73◊
4-1-1	.57	.53	.51	.54	.57	.54	.55	.55
4-2-2	.83♦	.75◊	.71◊	.73◊	.77◊	.65	.66	.72◊
4-3-1	.84♦	.77◊	.72◊	.73◊	.81♦	.66	.71◊	.77◊
5-4-1	.84♦	.79◊	.73◊	.74◊	.83♦	.67	.71◊	.77◊
6-4-2	.83♦	.79◊	.73◊	.73◊	.83♦	.67	.70◊	.77◊
6-5-1	.85♦	.80♦	.74◊	.74◊	.84♦	.67	.73◊	.79◊
7-6-1	.86♦	.87♦	.76◊	.83♦	.83♦	.70◊	.75◊	.80♦
8-7-1	.87♦	.89♦	.76◊	.86♦	.84♦	.71◊	.78◊	.82♦
◊'s count	2	7	**7**	8	2	**2**	**6**	6
♦'s count	**7**	**3**	0	**2**	**6**	0	0	**2**
Ability	XLNT	XLNT	AXPT	XLNT	XLNT	AXPT	AXPT	XLNT

Table 5.3. All the experiments with $AUC \geq 0.7$. (◊ : $0.7 \leq AUC < 0.8$, ♦ : $0.8 \leq AUC < 0.8$, XLNT: Excellent, AXPT: Acceptable.)

	Dataset A		Dataset B	
	model A	model B	model A	model B
bert-stratified	-	**Excellent**	-	Acceptable
bert-group	-	**Excellent**	-	-
tfidf-stratified	**Excellent**	Acceptable	**Excellent**	**Excellent**
tfidf-group	-	-	Acceptable	-

Table 5.4. The feature sets are acceptable and excellent in terms of description ability by considering the two best AUC scores of each feature.

Table 5.3 shows the AUC score of the feature sets which performs at least as well as "acceptable". Following Hosmer's general guidelines [18], we rank the feature sets' description ability in the table. The rank we used are Excellent (♦ : $0.8 \leq AUC < 0.8$) and Acceptable (◊ : $0.7 \leq AUC < 0.8$). One feature-split's ability will be ranked with the higher rank whose count greater than 1. For example:

- For Dataset A, tfidf-stratified with model A provides (◊'s count=2, ♦'s count=7). The higher rank with the count greater than 1 is ♦ (Excellent). So it is ranked as "Excellent".
- For Dataset B, tfidf-grouped with model A provides (◊'s count=2, ♦'s count=0). The higher rank with the count greater than 1 is ◊ (Acceptable). So it is ranked as "Acceptable".

To clearly show the different feature-splits' ability of description, we take the ability values in Table 5.3 and list them separately in Table 5.4. It shows "bert" feature shines on Dataset A, which is the much bigger one. And the "tfidf-stratified" generally perform well on the two datasets with both models.

	Dataset A			Dataset B		
	model A	model B	Difference	model A	model B	Difference
bert	.588	.749	-.161	.597	.654	-.056
liwc	.601	.556	.045	.622	.602	.020
lm	.647	.535	.112	.642	.552	**.089**
stat	.618	.525	.093	.567	.503	.064
tfidf	.729	.664	.065	.712	.675	.037
vader	.613	.491	**.123**	.604	.535	.067

Table 5.5. Average AUC difference between model A and model B with each feature set.

Additionally, comparing the model A and B, we noticed that model A with all the features outperform model B except with "bert". Table 5.5 shows this difference, where $Difference = model\ A's\ AUC - model\ B's\ AUC$. Each AUC score in this table is averaged over all the *n-p-st* combinations. "bert" is with the smallest difference (only negative value). In other words, among all the features, only "bert" feature with model B works better than model A. The possible reason is that with the "bert" feature the model only uses the sequence at maximum length of 512[2], then the concatenated period notes in model A hides up more information. In the meantime, "lm" and "vader" features yield the largest differences. It shows these features work better with longer text samples.

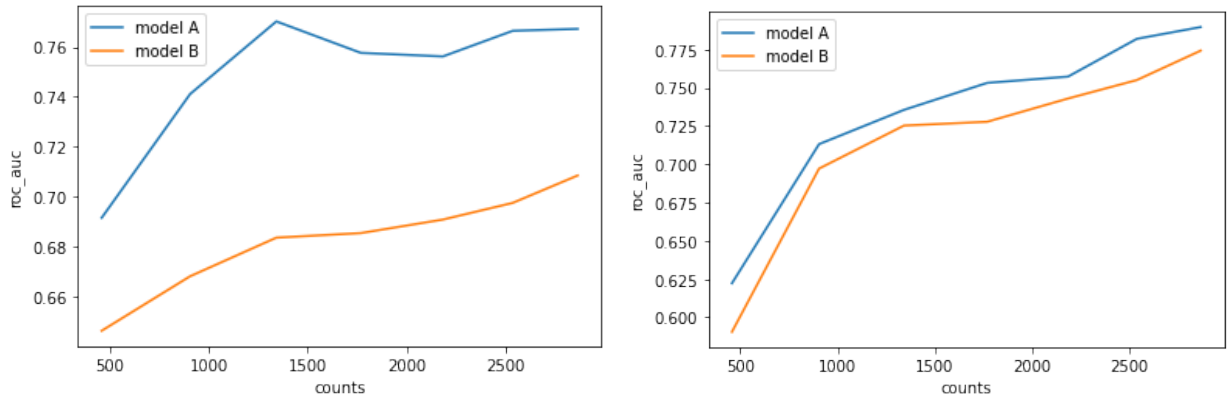


Fig. 5.7. Dataset A (left) and Dataset B (right): The growth of AUC score becomes slowly when the dataset count increases.

[2]This is due to the parameter of the pretrained BERT model we use.

Since the feature "tfidf" works the best among other features on Dataset B, we illustrate the growth trend of the AUC score with this feature when the counts of dataset increase. Fig 5.7 show the growth slows down on both datasets.

5.3. Task Pred(Class|employee)

Fig. 5.8 and Fig. 5.9 show the performance with different feature sets. Obviously, "tfidf" is the only feature which gives an "acceptable" description all the time. And the 4-4 yields the same performance as 3-3. At the mean time, 2-2, among the three $MID-FIN$ combinations, with "tfidf" is slightly better on Dataset A but slightly worse on Dataset B. The "bert" feature only shines with 4-4 on Dataset A.

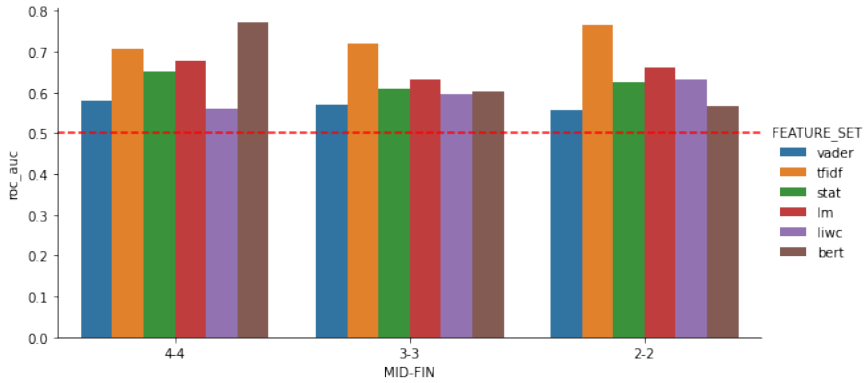


Fig. 5.8. Dataset A: AUC score on different $MID-FIN$ combinations

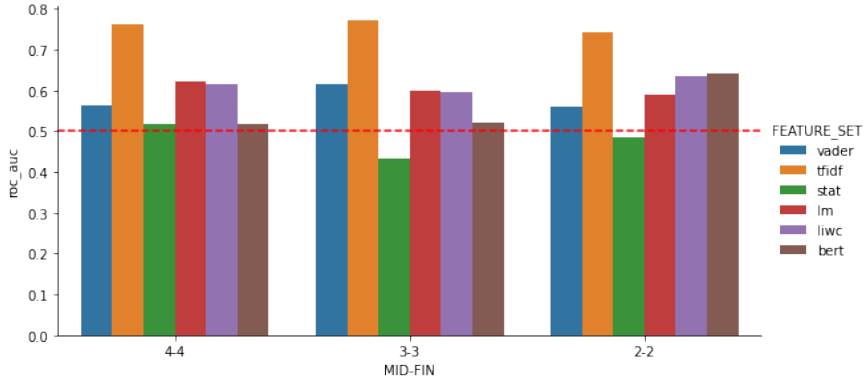


Fig. 5.9. Dataset B: AUC score on different $MID-FIN$ combinations

5.4. Conclusion of the experiment results

The feature "tfidf" is the only one that gives the description ability better than "acceptable" on both datasets for all the 3 tasks. It shows "excellent" description ability in Task Pred(Class|note) and Pred(Class|period), and "acceptable" description ability in Task

Pred(Class|employee). The feature "bert" gives basically "excellent" on dataset A and "acceptable" in Task Pred(Class|note) and Pred(Class|period), but very low performance in Task Task Pred(Class|employee). Those conventional emotional analysis features such like VADER or LIWC are far behind. It shows that in this kind of professional progress notes, emotion detection is a hard job. It also show that the classification of the notes or periods of notes are easier than the classification of the employees. A system built on the notes wise or periods of notes wise will not only have a finer granularity and accuracy but also have more training samples available.

In addition, in terms of the classifying a user's notes, training the model with a part of this user self's note ("stratified" splits) always gives much better performance than training solely with other users' notes ("group" splits). It implies the temporal classification model should be considered in the future research. It will help us train the model do the prediction based on user's historic note, which will avoid training on the "new" data and predicting the "old" data.

Chapter 6

Analysis

In this chapter, coefficients of the TFIDF features of the models trained in Task $Pred(Class|note)$ will be interpreted. According to Table 5.2, the models with TFIDF feature and "stratified" split method are providing excellent discrimination on both datasets. We draw the conclusion based on the trained models rather than about the true (real-world) generative process of the data.

6.1. Coefficient variety analysis

We use the logistic regression classifier as $h(\boldsymbol{x})$ in this research. Given a note's vector representation $\boldsymbol{x} = \{x_1, \ldots, x_d\}$, the probability of $Y = 1$ is denoted as $p = P(Y = 1)$.

$$p = S(\beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d) \tag{6.1.1}$$

Where the $\beta_0, \ldots, \beta_d$ are the regression coefficients and S is the e based sigmoid function. The binary label $\hat{y}$ predicted by the model is defined in Equation 3.1.2 and extended as:

$$\begin{aligned} \hat{y} &= h(\boldsymbol{x}) \\ &= \begin{cases} 1 & \text{if } p \geq 0.5, \\ 0 & \text{if } p < 0.5. \end{cases} \end{aligned} \tag{6.1.2}$$

Specifically, when the note is presented with a TFIDF standardized[1] vector x, these coefficients will tell us how does the model think some TFIDF features are more important than others. We take a simple case as an example. Suppose there are two TFIDF features: "bad" and "normal", the x_{tfidf_bad} and x_{tfidf_normal} are the standardized vector presentation for the two features respectively.

[1]The standard score of a sample x' is calculated as:

$$x = \frac{(x' - u)}{s}$$

Where u is the mean and s is the standard deviation of the population.

$$p = S(0.2 \cdot x_{tfidf\_bad} + 0.02 \cdot x_{tfidf\_normal})$$
$$= \frac{1}{1 + e^{-(0.2 \cdot x_{tfidf\_bad} + 0.02 \cdot x_{tfidf\_normal})}}$$

Then the log-odds of p is:

$$\ln \frac{p}{1-p} = 0.2 \cdot x_{tfidf\_bad} + 0.02 \cdot x_{tfidf\_normal}$$

- It means increasing the x_{tfidf_bad} by 1 increases the log-odds by 0.2, the odds that $Y = 1$ increase by a factor of $e^{0.2} \approx 1.22$.
- It means increasing the x_{tfidf_normal} by 1 increases the log-odds by 0.02, the odds that $Y = 1$ increase by a factor of $e^{0.02} \approx 1.02$.

Observing the value of coefficients, we conclude that the TFIDF feature "bad" is more important than "normal" in this case.

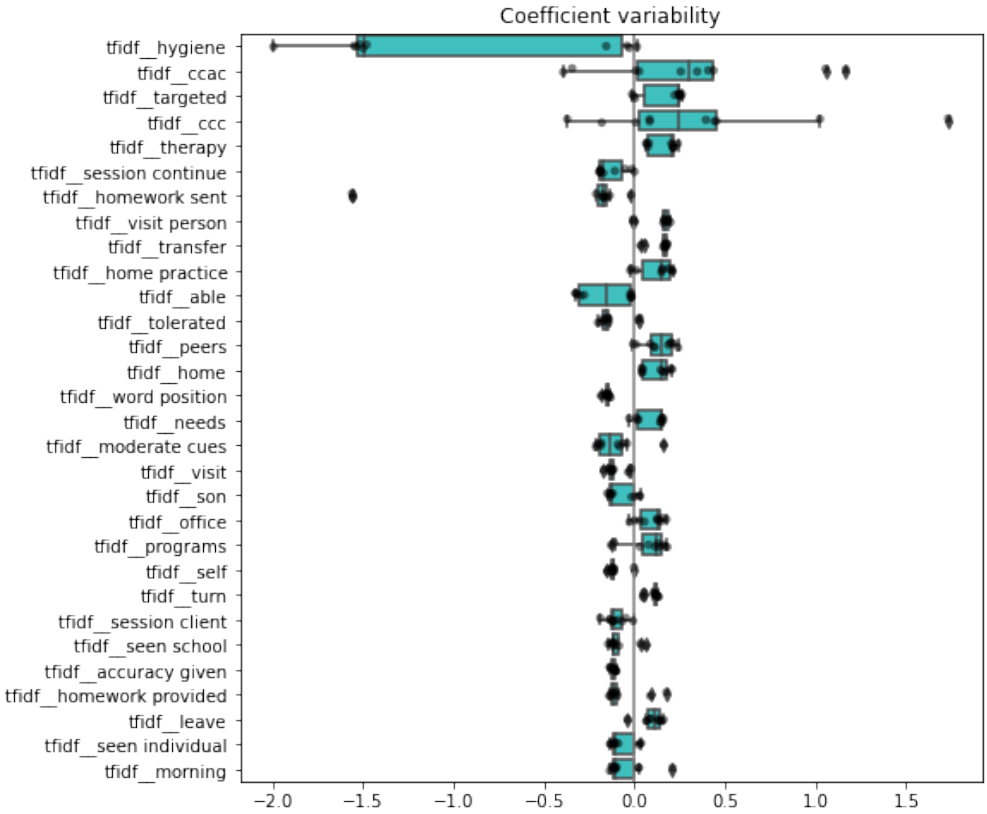


Fig. 6.1. Dataset A: Coefficient variability Top 30 through cross-validation ($k = 10$). The coefficients of the TFIDF features of the models trained in Task $Pred(Class|note)$ are analyzed.

However, we should be prudent to interpret the coefficients which vary significantly [38]. For this reason, the coefficient variability through cross-validation will be examined. In addition, the 25th percentile (Q1) and the 75th percentile (Q3) of a coefficient's distribution

should be both negative or positive. This will make sure the corresponding TFIDF features contribute to the positive label or negative label at least 75% of all the cases.

For Dataset A, as shown in Fig. 6.1, among the top 30, there are 14 coefficients on the positive side, which we believe are important for this model to predict the positive labels. The corresponding TFIDF features are[2]:

'ccac', 'targeted', 'ccc', 'therapy', 'visit person', 'transfer', 'turn', 'home practice', 'peers', 'home', 'needs', 'office', 'programs', 'leave'

- Note id: 590
Late entry - February 8th 2017 Block 1, visit #7 S: Picked up client from class. Motivated to participate. Hands sanitized before and after session. Homework returned. O/A: Targeted /s/ and /z/ at structured conversation: initial: 83% medial: 68% - difficulties when /s/ is in cluster eg. loST, missed [mIst], juST final: 75% vocalic /l/ at structured conversation: 75% Multisyllabic words at connected speech: 3 syllables: 83% 4 syllables: 83% 5 syllables: 75% with moderate cues Continued to incorporate language strategies as outlined in previous progress note. P: Continue goal directed tx. Homework provided.
- Note id: 599
February 9, 2017 B2 V8 S: Client picked up from classroom. Pleasant and motivated to participate. O/A: Targeted [ch] at structured conversation: Initial: 74%, medial: 85%, final: 79% [S] at sentence level with mod-max cueing: initial: 85%, medial: 84%, final: 88% (plan to target to structured conversation) P: Continue goal directed tx. Multisyllabic words (3-4 syllables) were noted to adequate at sentence level and therefore not targeted at structured conversation. Homework provided. Feb 16th 2017 Attempted to pick up student from class but teacher said he was writing a French test. Writer not able to switch students around. Plan to return following week. Notified mother who said, if possible, next time, can CLIENT write at another time for tests. Plan to inform teacher.
- Note id: 995
Monday June 9th, 2014 Visit #8 [PERSON] [PERSON] is trying to use his speech sounds at recess with one friend, he said it went 'kind of ok.' but that it is was weird. He will continue to keep trying and understands why it is so important. 'or' medial carryover sentences 76% /s/ and /z/ need work. /ar/ medial sentences-fill in the blank, 70%. 'sh' has decreased rounding at times--this was targeted last year. 'er' weak inconsitent though. Father to set more goals with him for the summer--camp speech goals. Good progress. Practices daily with Dad. Janna Adler, M.Sc., SLP, reg CASLPO 3570

Fig. 6.2. Some note samples that contain "targeted".

We manually go through the notes and find the usage of "targeted" is strongly related to the work content of speech therapy. Some note samples are shown in Fig. 6.2 as examples. In these examples, the object of "targeted" are /s/, /z/, structured conversation and /sh/. These sound symbols and professional terms are commonly used by speech therapists in the description of job schedule or progress.

To get more information about the context of each feature, we examined the left and right context (2-words[3]) of each feature. This is a heuristic choice. We created the 1-word, 2-words, and 3-words context for some selected features and compared the expression of context with different length. The 2-words context is generally good enough to present the context of the features. For example, the Fig. 6.3 shows the top 20 left and right context of the feature 'targeted'. The total count of its Top 20 left context is 2743 and the total count of its Top 20 left context is 8849. It gives the coverage of 31%. Similarly, its Top 20 right context gives the coverage of 24%. In the left context, according to the words like "cooperative", "medial"

[2] $[PERSON]$ is introduced into the data as the placeholder of a person's name during the anonymization.
[3] It means, for one-word feature, the 3-grams is used; for two-word feature, the 4-grams is used.

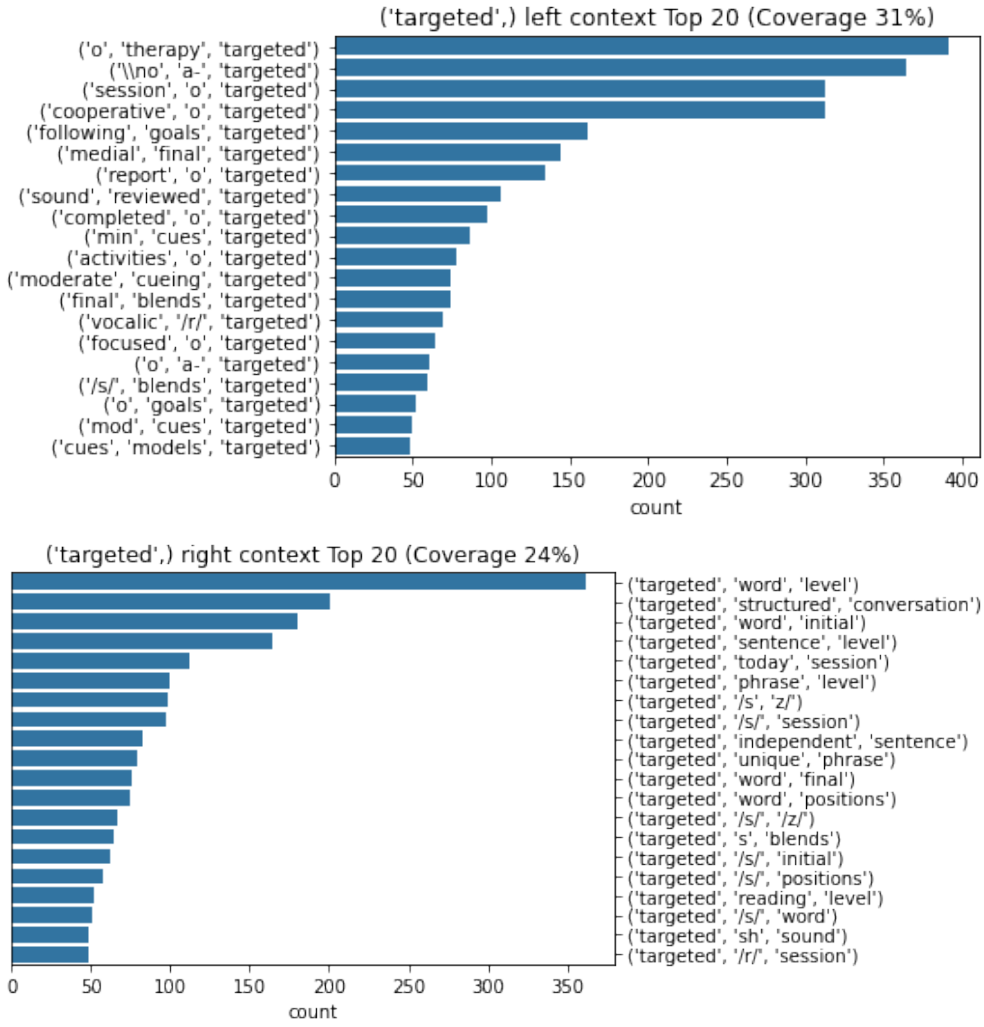


Fig. 6.3. The left and right context of 'targeted' for Dataset A.

and "sound", it shows that "targeted" relates to the work content of speech therapy. In the right context, "word", "level", "position" and different phonetic symbols confirms that this feature relates to the work content of speech therapy. We don't think this feature directly relate to the dissatisfaction. The reason that this feature contributes to the prediction of the positive case is due to the work content due to our analysis. For example, many speech therapists may choose to turn over after they finish the current treatment session of the patient. During the end of their session, they may mention the term "targeted" more often to optimize their treatment achievement. For this reason, eventually, we categorize the feature as "work content".

Take the term "leave" as another example. Some note samples are shown in Fig. 6.4 as examples. In these examples, there are generally two kinds of usage of "leave". In the first case, as in note id 320 and 87166, it is used as a verb meaning deliver. If the caregiver

- Note id: 320
This client showed up to their appointment on Friday January 8th, 2016, and were very upset in the matter that they did not receive a phone call to notify them. However I called the family three times and emailed them. There was no voicemail to leave a message. Anyways they told Jean they do [PERSON] want to be contacted in the future and they are done with services. I just tired calling and someone hung up on me. This client has been discharged from the program.
- Note id: 87166
Tried calling mother to confirm email address as the emails were bouncing back. Was unable to leave a voice message on mother's cell. Called father and he informed that a family emergency had come up and mom is now in Holland. She will be returning December 5th, 2018. Will call to rebook the consult around then.
- Note id: 14994
October 29, 2018. Received email from father that Maliha will be on a [PERSON] trip tomorrow. Responded that SLP will have approx 1 more visit with Maliha before maternity leave begins, [PERSON] to resume in January with a new clinician. Tomorrow\'s session postponed.
- Note id: 15006
October 29, 2018. Note sent home regarding SLP\'s upcoming maternity leave at end of November - therapy to resume in January with new clinician.

Fig. 6.4. Some note samples that contain "leave"

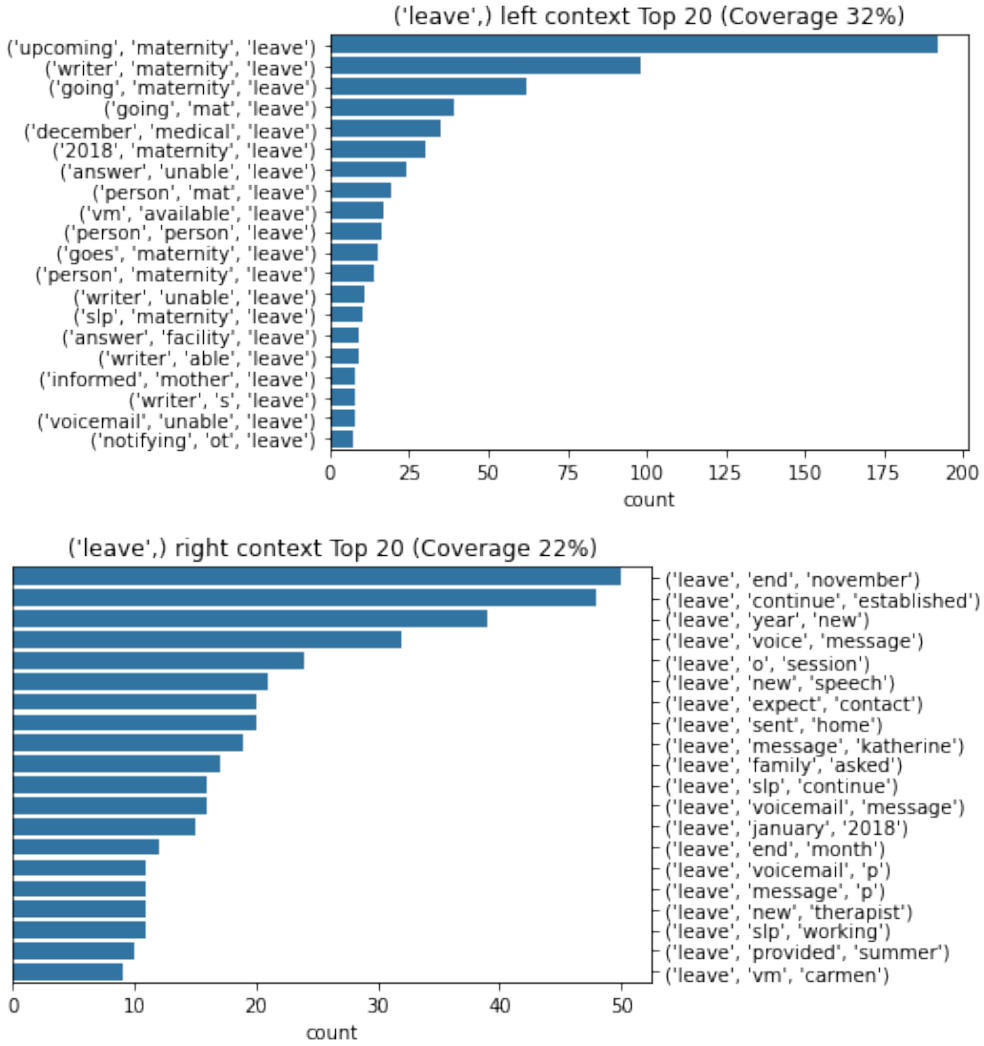


Fig. 6.5. The left and right context of 'leave' for Dataset A.

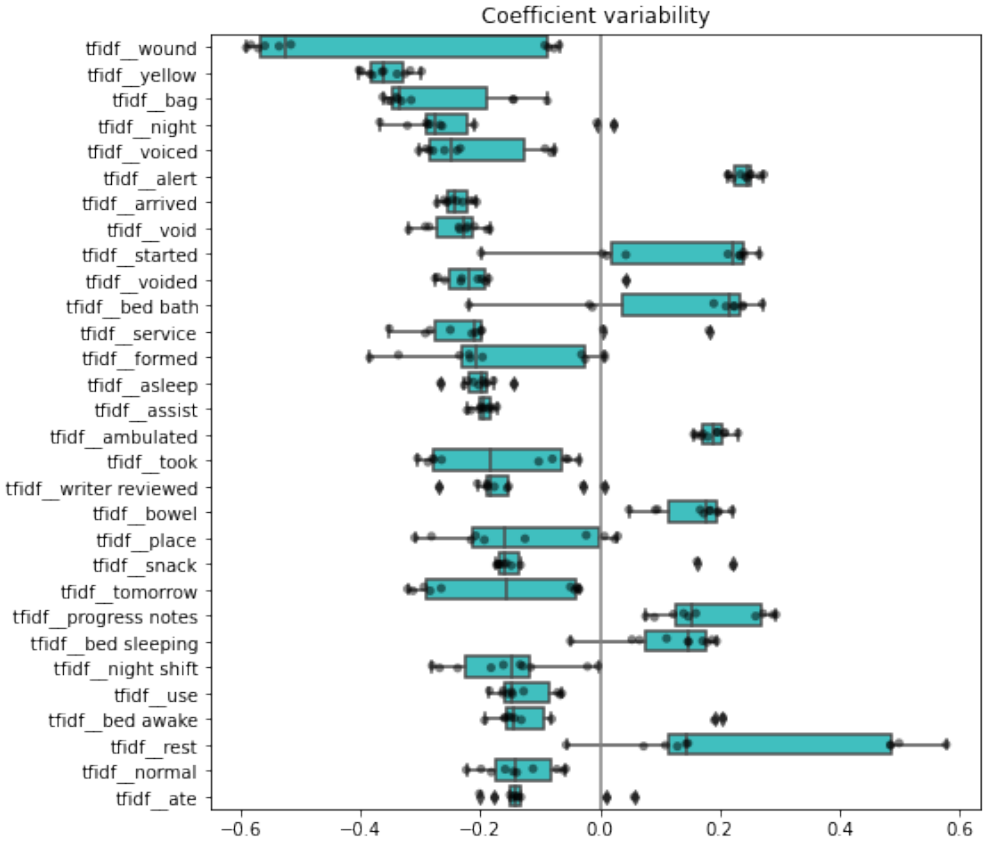


Fig. 6.6. Dataset B: Coefficient variability Top 30 through cross-validation ($k = 10$). The coefficients of the TFIDF features of the models trained in Task $Pred(Class|note)$ are analyzed.

can't contact clients, there's no doubt it will cause the communication barrier and the dissatisfaction [29]. In the second case, as in note id 14994 and note 15006, it is used as a verb meaning depart. Either the caregiver's or other peers' depart will be mentioned in the note. We suspect, if other peers are taking the leave, the caregiver will have to take more workload. For this reason, it may bring up dissatisfaction as well. To get more information about the context of this feature, the Fig. 6.5 shows the top 20 left and right context of the feature 'leave'. In the left context, according to the words like "maternity" and "medical", it shows that "leave" relates to the second case we manually analyzed. In the right context, the word "message", "voice", "voicemail" and "contacted" shows "leave" relates to the first case. So we categorize the feature as "work environment" and "Communication". Similarly, for Dataset B, as shown in Fig. 6.6, there are 8 coefficients on the positive side among the top 30. Among these 8, the feature "progress note" seems special. Even though there are around one thousand notes containing the "progress note", but most of them are quite short as show in Fig. 6.7. The left and right context of the feature "progress note" are shown in Fig. 6.8. The high coverage in this figure shows the homogeneity of the context. We strongly suspect

- Note id: 405
Author reviewed `progress notes` .
- Note id: 436
Writer reviewed the `progress notes` .
- Note id: 4494
Writer reviewed `Progress Notes` .
- Note id: 4497
Writer reviewed `Progress Notes` and Service Tasks.
- Note id: 5995
`Progress notes` and ADLs reviewed by writer.
- Note id: 6000
`Progress notes` and ADLs reviewed by writer.

Fig. 6.7. Some note samples that contain "progress notes"

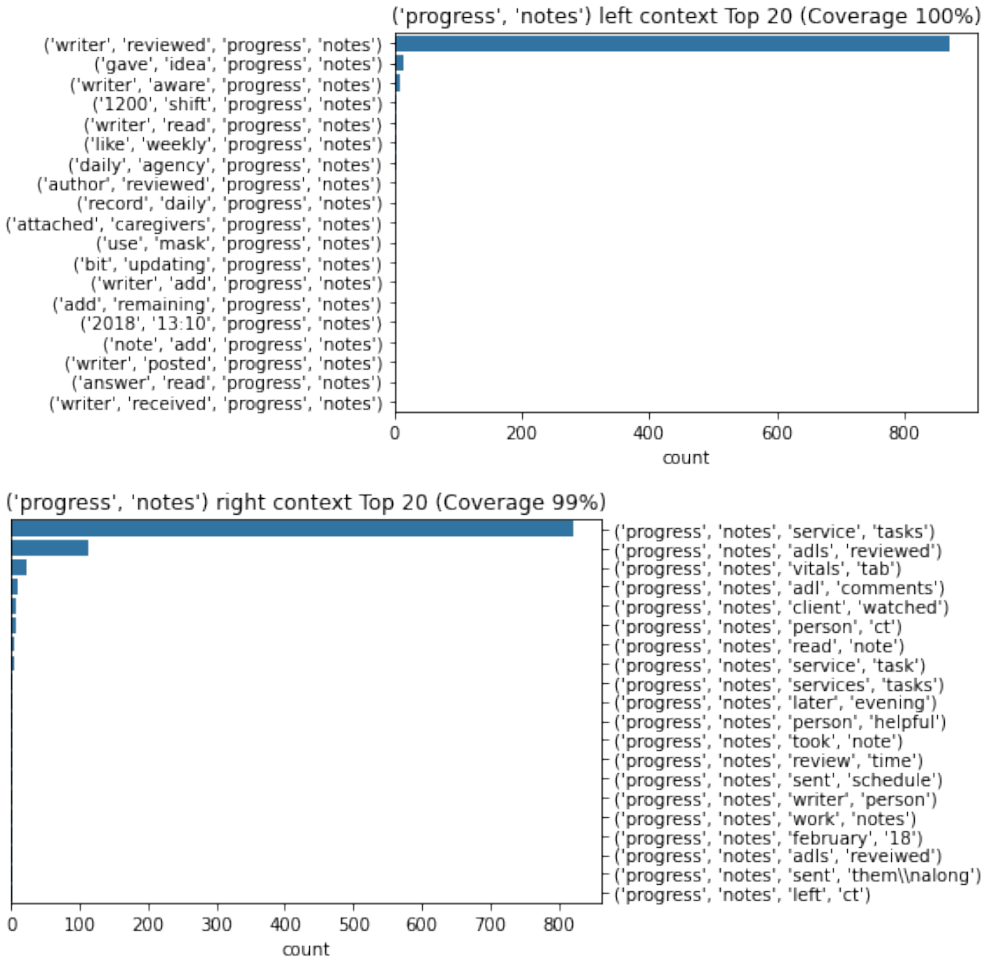


Fig. 6.8. The left and right context of 'progress note' for Dataset B.

that this feature related to the caregiver's dissatisfaction. The notes are quite short and there's no concrete work content details presented.

We categorized manually 14 corresponding TFIDF features for Dataset A, and 8 corresponding TFIDF features for Dataset B. As shown in Table 6.1, most of the features are

Category	Dataset A	Dataset B
Work content (patient status)	'targeted', 'therapy', 'home', 'home practice', 'needs', 'programs', 'transfer', 'peers', 'visit person', 'visit', 'turn', 'office','ccc', 'ccac'	'alert', 'ambulated', 'bed bath', 'bed sleeping', 'bowel', 'rest', 'started', 'progress note'
Work environment (administrative management, colleague relationship)	'leave'	'progress note'

Table 6.1. Important TFIDF features for both datasets.

categorized as 'work content'. The features 'leave' and 'progress notes' are categorized as 'work environment'. Their context are shown in Fig. 6.5 and Fig. 6.8 respectively. As we analyzed, the "leave" usage probably shows two situations:

- The caregiver can't contact clients;
- The caregiver's or other peers' depart.

The "progress note" usage shows the situation that there's no concrete work content details presented. For this reason, these 'work environment' related factors will probably cause the dissatisfaction.

6.2. Clustering of important features by K-means

In the last section, we presented the categorized results that we got by examining these important features manually. In this section, we will show the clustering completed by the algorithm. The generation of the context vectors will be explained. Then, the application of the Principle Component Analysis (PCA) to reduce the dimensionality of these context vectors will be presented. Lastly, the clustering results will be shown by applying the K-means method.

6.2.1. Features' Term-Term Matrix

In Section 6.1, we spotted 14 important features from Dataset A and 8 important features from Dataset B. We will create the vectors for each of them by counting the word occurrence in their left and right context. Each feature's context forms a document. We denote a feature's context as d. The score $f_{t,d}$ is the number of times that term t occurs in context d (l words to the left and l words to the right). It could be seen as "features' feature". We define our corpus as $D = \{D_a,D_b\}$, where $D_a = \{d_1,d_2,...,d_{14}\}$ is the corpus of 14 features for Dataset A and $D_b = \{d_{15},d_{16},...,d_{22}\}$ is the corpus of 8 features for Dataset B. The feature dimension of each d_i is $|\{V_a,V_b\}|$, with $V_a = \{v_1,v_2,...,v_p\}$, $v \in d$, $d \in D_a$ and

$V_b = \{v_1, v_2, ..., v_q\}$, $v \in d$, $d \in D_b$. We are actually putting the feature vectors of the 14 features from the Dataset A and the ones of 8 features from Dataset B into the same linear space.

In our case, we set $l = 2$, which means the context window is 2 words to the left and 2 words to the right. It gives us $|\{V_a, V_b\}| = 12790$. In this way, our term-term matrix dimension for our 22 features are 22 by 12790.

6.2.2. Dimension reduction with PCA

The dimension of the features is too large compared to the number of samples. In addition, our 22 by 12790 term-term matrix is quite sparse. We choose the PCA [32] method to reduce the dimension. Before applying the PCA, we standardize the matrix by removing the mean and dividing by the standard deviation.

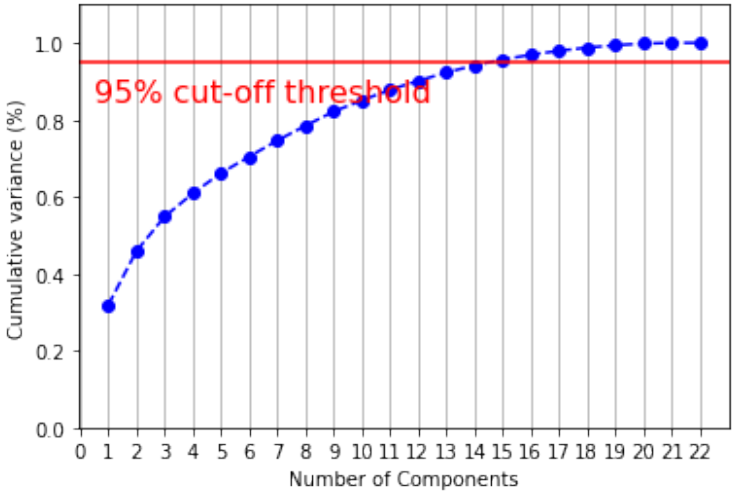


Fig. 6.9. Cumulative variance on different number of components in PCA

Choosing the number of principal components (PCs) in PCA is usually heuristic. We choose to keep 95% variance of the matrix and present the cumulative variance in Fig. 6.9. It shows that 15 PCs will keep the percentage of variance we chose. The dimension reduced features' term-term matrix is 22 by 15 therefore.

We present the visualization based on 2PCs and 3PCs in Fig. 6.10. In the 2D situation, the "therapy" and "home" are far away from the others. In the 3D situation, the "therapy", "home" and "needs" are far from the others. Nevertheless, the visualization doesn't show not much details in this case due to the lack of specificity for the majority.

A heat map of the correlation matrix of the dimension reduced features' term-term matrix is shown in Fig. 6.11. Generally, it shows the features from Dataset B correlate with each other. It also shows interestingly some features from dataset A (e.g., "ccc", "visit person", "leave") correlate to the features from dataset B.

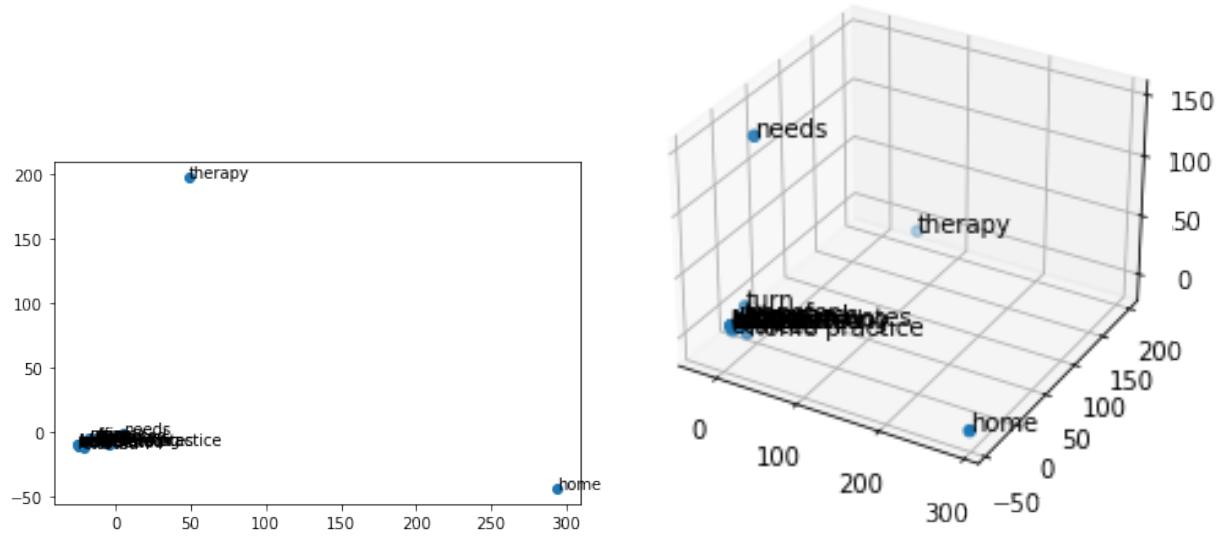


Fig. 6.10. Visualization of the dimension reduced features' term-term matrix (2PCs on the left and 3PCs on the right).

Fig. 6.11. Heat map of the correlation matrix of the dimension reduced features' term-term matrix

6.2.3. Clustering

The intuition is to separate the features those causing the dissatisfaction from the features solely relating to the job content.

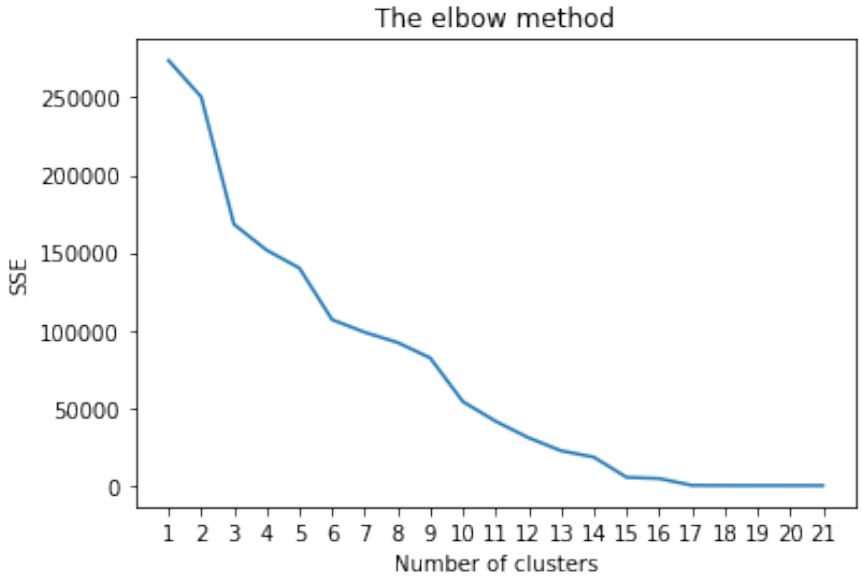


Fig. 6.12. Elbow method for choosing the number of clusters in K-means

The elbow method is usually used to decide the number of clusters in K-means [25]. The basic idea of this method is that calculating the sum of Euclidean distance between the data points and their respective cluster centroids on different number of clusters. Then we need to find the "elbow" point that after which the SSE decreases relatively slowly as the number of clusters increases. Nevertheless, in our case, there is no clear "elbow" point as shown in Fig.6.12. We choose 10 clusters based on our observation of the clustering result.

Cluster id	Features in Dataset A	Features in Dataset B
1	'ccac', 'ccc', 'visit person', 'home practice', 'office', 'leave',	'alert', 'bed bath', 'ambulated', 'bowel', 'progress notes', 'bed sleeping', 'rest'
2	'home'	-
3	'therapy'	-
4	'needs'	-
5	'targeted'	-
6	'turn'	-
7	'programs'	-
8	'peers'	-
9	'transfer'	-
10	-	'started'

Table 6.2. Clustering result by K-means with 10 clusters

The clustering results by K-means with 10 clusters are shown in Table 6.2. In cluster 1, there are 6 features from dataset A and 7 features from dataset B. In other clusters, there is only 1 feature in each of them. Nevertheless, according to our observation, non of the cluster of features relate generally to the caregivers' dissatisfaction.

6.3. Conclusion of the analysis

The section shows that manual method outperforms the clustering algorithm. It also shows that our model is able to help us find the dissatisfaction factors even though the factors are quite few.

Chapter 7

Conclusion

The detection of the home healthcare worker's dissatisfaction is defined as a supervised binary classification problem in this research. We conducted the experience on real data collected by two AlayaCare's tenants. During the exploration of the nature of the data, we illustrated the number of notes by user, the number of notes by week and the distribution of the length of note time span in Chapter 2. It shows the two data sets are similar to each other in terms of the distribution of number of notes and the length of notes.

The input is generally a progress note (or a bunch of concatenated notes) coming from one healthcare worker, and the output is the prediction over one of the binary value positive or negative, which correspond to healthcare worker's job post status "terminated" and "active" respectively. We designed three tasks with three different granularities of the input and output. A single note, a period of notes and a historic/recent period combination of notes from a worker are used as three different kinds of input in our three tasks. The class of a note, class of a period and class of an employee are predicted.

The effect of performance by the balance of positive and negative data interests us. We carefully engineered these tasks paying attention to data balance. By varying our labeling related hyper-parameters, the balanced-data situations and the imbalanced-data situation are covered in Task Pred(Class|note) and Task Pred(Class|period). The results show that generally, a balanced input outperforms an imbalanced one. We also compared the models with different features fed into a logistic regression classifier, including TFIDF, BERT, VADER, LIWIC, statistical features and language model features.

Applying the "double-layer" cross-validation, as discussed in Chapter 3, we tuned the hyper-parameters with first cross-validation and trained our models with second cross-validation to overcome the overfiting issue. The models with TFIDF and BERT features showed excellent description ability in two tasks out of three as shown in Chapter 5. However the TFIDF feature works mostly the best for all the tasks due to its large dimension and the capability of taking the whole sample into account compared to the BERT feature

which only handles up to 512 tokens. By analyzing the coefficient variability of our regression model with TFIDF feature, we identified the "work environment" related dissatisfaction factors in Chapter 6. Our designed combinations of labeling hyper-parameters also give us a chance to observe the relationship between the performance and the number of samples. The results show that expectedly the experiments trained with more samples generally give a better performance.

Among the three tasks we designed, Task Pred(Class|note) and Task Pred(Class|period) are relatively easier than Task(Class|employee) which means predicting the class of a note or a period of note are easier than predicting the class of an employee. It makes sense that:

(1) The sample numbers in Task Pred(Class|employee) is quite small;

(2) The employees write the notes which include the dissatisfaction factors don't necessarily turnover or turnover immediately in real world.

From the medical point of view, dissatisfaction is not like depression which is a disease and there is a standard about how to diagnosis it. However, Our research shows detecting the dissatisfaction factors from caregiver's note is possible.

Following this research, there are many paths this domain could follow, some of them being listed here:

(1) A temporal classification model could be applied;

(2) BERT features could be used on larger datasets to prove its ability;

(3) An end2end BERT solution should be designed, deal with the fact that data is rather specific (challenging for fine tuning the model) and the notes maybe rather long (challenging long dependency issues)

(4) Some domain oriented linguistic features could be explored with the cooperation of the domain professionals;

(5) In order to have a common benchmark for researchers, the anonymization of a progress note dataset could be done, allowing to share it ethically.

References

[1] Hertz A. and Lahrichi N. A patient assignment algorithm for home care services. *Journal of the operational research society*, 60(4):481–495, 2009.

[2] Pang B. and Lee L. Opinion Mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(ue 1-2):1–135, 2008.

[3] Majid Bagheri Hosseinabadi, Mohammad Hossein Ebrahimi, Narges Khanjani, Jamal Biganeh, Somaye Mohammadi, and Mazaher Abdolahfard. The effects of amplitude and stability of circadian rhythm and occupational stress on burnout syndrome and job dissatisfaction among irregular shift working nurses. *Journal of Clinical Nursing*, 28(9-10):1868–1878, May 2019.

[4] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997.

[5] David Goldberg and Nohel Zaman. Text Analytics for Employee Dissatisfaction in Human Resources Management. In *Twenty-fourth Americas Conference on Information Systems*, New Orleans, 2018.

[6] Munmun De Choudhury and Scott Counts. Understanding affect in the workplace via social media. In *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*, page 303, San Antonio, Texas, USA, 2013. ACM Press.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[8] Chang E., Cohen J., Koethe B., Smith K., and Bir A. Measuring job satisfaction among healthcare staff in the United States: a confirmatory factor analysis of the Satisfaction of Employees in Health Care (SEHC) survey. *International Journal for Quality in Health Care*, 29(2):262–268, 2017.

[9] David A. Etzioni, Jerome H. Liu, Melinda A. Maggard, and Clifford Y. Ko. The Aging Population and Its Impact on the Surgery Workforce. *Annals of Surgery*, 238(2):170–177, August 2003.

[10] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[11] C. Fikar and P. Hirsch. Home health care routing and scheduling: A review. *Computers & Operations Research*, 77:86–95, 2017.

[12] Arthur Marçal Flores, Matheus Camasmie Pavan, and Ivandré Paraboni. User profiling and satisfaction inference in public information access services. *Journal of Intelligent Information Systems*, August 2021.

[13] J Forster and B Entrup. A Cognitive Computing Approach for Classification of Complaints in the Insurance Industry. *IOP Conference Series: Materials Science and Engineering*, 261:012016, October 2017.

[14] Coppersmith G., Dredze M., Harman C., Hollingshead K., and Mitchell M. CLPsych 2015 shared task: Depression and PTSD on Twitter. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 31–39, 2015.

[15] Louis Hickman, Koustuv Saha, Munmun De Choudhury, and Louis Tay. Automated tracking of components of job satisfaction via text mining of twitter data. In *ML Symposium, SIOP*, pages 1–11, 2019.

[16] Robert Holly. 2020 Home-Based Care Technology Survey, 2020. Available at https://f.hubspotusercontent00.net/hubfs/2702101/HHCNAlayaCareSurvey.pdf.

[17] Robert Holly. 2021 Home Care Employee Retention Survey Report, 2021. Available at https://f.hubspotusercontent00.net/hubfs/2702101/HHCN_Survey_AlayaCare_2021-Home-care-employee-retention_FINAL.pdf.

[18] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. ch5 Assessing the Fit of the Model. In *Applied logistic regression*, number 398 in Wiley series in probability and statistics, page 177. Wiley, Hoboken, New Jersey, third edition edition, January 2013.

[19] C J Hutto and Eric Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, page 10, 2015.

[20] Wolohan J.T., Hiraga M., Mukherjee A., and Sayyed Z.A. Detecting Linguistic Traces of Depression in Topic-Restricted Text: Attending to Self-Stigmatized Depression with NLP. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 11–21, 2018.

[21] Yeonjae Jung and Yongmoo Suh. Mining the voice of employees: A text mining approach to identifying and analyzing job satisfaction factors from online employee reviews. *Decision Support Systems*, 123:113074, August 2019.

[22] Vincent Kriz, Martin Holub, and Pavel Pecina. Feature Extraction for Native Language Identification Using Language Modeling. *RANLP*, page 9, 2015.

[23] Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models.

Journal of Cheminformatics, 6(1), December 2014.

[24] Leodoro J. Labrague and Janet Alexis A. Santos. Fear of COVID-19, psychological distress, work satisfaction and turnover intention among frontline nurses. *Journal of Nursing Management*, 29(3):395–403, April 2021.

[25] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[26] Cissé M., Yalcindag S., Kergosien Y., Sahin E., Lenté C., and Matta A. OR problems related to home health care: A review of relevant routing and scheduling problems. *Operations Research for Health Care*, 14:1–22, 2017.

[27] De Choudhury M., Gamon M., Counts S., and Horvitz E. Predicting Depression via Social Media, Seventh Int. *AAAI Conf. Weblogs Soc. Media*, 2:128–137, 2013.

[28] Sargen M., Hooker R.S., and Cooper R.A. Gaps in the supply of physicians, advance practice nurses, and physician assistants. *J Am Coll Surg*, 212:991–999, 2011.

[29] Donna K. McNeese-Smith. A content analysis of staff nurse descriptions of job satisfaction and dissatisfaction. *Journal of Advanced Nursing*, 29(6):1332–1341, June 1999.

[30] McHugh M.D., Kutney-Lee A., and Cimiotti J.P. Nurses' widespread job dissatisfaction, burnout, and frustration with health benefits signal problems for patient care, 2011.

[31] Raghavendra Pappagari, Piotr Zelasko, Jesus Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844, SG, Singapore, December 2019. IEEE.

[32] Karl Pearson. LIII. *On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, November 1901.

[33] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The Development and Psychometric Properties of LIWC2015. *UT Faculty/Researcher Works*, page 26, September 2015.

[34] Home Care Pulse. 2021 Home Care Benchmarking Study, 2021.

[35] Muhammad Saqlain Rehan, Furqan Rustam, Saleem Ullah, Safdar Hussain, Arif Mehmood, and Gyu Sang Choi. Employees reviews classification and evaluation (ERCE) model using supervised machine learning approaches. *Journal of Ambient Intelligence and Humanized Computing*, May 2021.

[36] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*, February 2020. arXiv: 1910.01108.

[37] Guntuku S.C., Yaden D.B., Kern M.L., Ungar L.H., and Eichstaedt J.C. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.

[38] scikit-learn. Common pitfalls in interpretation of coefficients of linear models, March 2020.

[39] M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

[40] Bodenheimer T. and Sinsky C. From triple to quadruple aim: care of the patient requires care of the provider. *Ann Fam Med*, 12:573–576, 2014.

[41] K. Vangel. *Employee Responses to Job Dissatisfaction*. University of Rhode Island DigitalCommons@URI, 2011.

[42] WHO. Ageing and health, February 2018. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health.

[43] Tausczik Y.R. and Pennebaker J. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29:24–54, 2010.

[44] Isik Urla Zeytinoglu. The Impact of Implementing Managed Competition on Home Care Workers' Turnover Decisions. *Healthcare Policy = Politiques De Sante*, 1(4):106–123, May 2006. L'incidence de la mise en oeuvre d'une concurrence dirigée sur les décisions d'emploi des travailleurs des soins à domicile.