Final Report

Team 4
2013150035 Song Min
2014150023 Cho Youngsang
2015150007 Kang Hoseok
2017190034 Lee Eunjin
2017150418 Lee Nahyeon

1. Introduction of data

This Employment Scam Aegean Dataset (EMSCAD) is a set of data collected to conduct research on Employment Scam problem. We downloaded this data from Kaggle. This data was created by collecting job ads investigated between 2012 and 2014, and out of about 18,000 data, about 800 fake ones are mixed. There is a total of 18 variables in this data, and we divided the data into the four types: Binary, Category, Text and Complex. Complex type includes the variables that needs variable split since they includes several information.

Variables		
Name	Type	Details : number of missing values
Job_ID	Integer	Serial numbers of job posting
title	Text	Title of the job posting: 0
Location	Complex	Location, separated by the commas : 344
Department	Category	Departments, unique(1338), : 11516
salary_range	Complex	Range of salary, separated by the bars : 14971
company_profile	Text	Profiles, unique(1826) : 3301
description	Text	Description : 0
requirements	Text	Requirements : 2664

benefits	Text	Benefits of the job : 7010
telecommuting	Binary	Telecommuting or not : 12
has_company_logo	Binary	Has logo or not : 17
has_questions	Binary	Has screening question or not : 37
employment_type	Category	Type of employment : 3431,
required_experience	Category	Required experiences : 7938
required_education	Category	Required education level : 6937
industry	Category	Industry type : 4859
function.	Category	Function of job : 6382
fraudulent	Binary	Fraud posting or not : 46

[Table 1.1] Introduction of the variables

This data mainly consists of meta and text variables rather than typical continuous and categorical variables. Target variable is 'fraudulent', and it is binary, containing 0 and 1. There are many missing values, which can be a problem in general data analysis. We can consider the possibility that this data could be used as a feature. Our goal was developing a model that can detect the fake job posting through analyzing this data.

2. Preprocessing

1) Binary Data

There are three binary variables: telecommuting, has_company_logo, and has_questions. In case of binary data, there was no missing values. All of the values in these binary variables are 0 or 1, so we did not any additional preprocessing.

2) Categorical Data

There are six categorical variables: employment_type, required_experience, required_education, departments, industry and function. In case of employment_type, the percentage of missing value was 19.4%. We thought that this is not ignorable percentage, so we made another binary variable that implies whether this row has missing value on employment_type or not. There were five employment types: Full-time, Contract, Part-time, Temporary and Other.

Required_experience had 39% of missing values, so we made the binary variable as well. There were 8 kinds of categories in this variable. We did the same thing for requited_education, which contains 13 categories.

In case of department, industry and function, there were a lot of categories. Especially, in case of department, there were 1337 categories. There are 64% of missing values, and each categories takes up such a small partitions. In fake data, IT department had the highest percentage, but the missing value was much higher than that. Therefore, we concluded that we cannot use the department variable itself.

Industry had 131 categories, and 27% of them are missing values. Although there were less categories than the department, still, most of the categories had a small percentage. It was same for the function variable. What's more, there was no detailed difference between those three rows.

For example, some of the company wrote same word “Marketing” for all of the categories. Thus, we decided to combine the department, industry and function together and analyze it as a text dataset.

3) Text Data

Text data has five variables – title, company_profile, description, requirements and benefits. Title does not have any missing values. For three text variables(except title and description variable), we made the binary variables for each to indicate whether this company has missing value for each variable or not. Company_profile has 18% of missing value rate, about 0% for description, 15% for requirements, 40% for benefits and 0% for title.

Before starting the analyze, we can find out that there are some stopwords, which does not affect to the meanings: like ‘and’ and ’the’. We should erase those words, and we did it through using ‘nltk’ library in python.

After that, we used the stemming algorithms to find out the topic words and trim the words shorter. These erase the suffix from the word, and it makes it easier to find out the core of the word one by one. We used two kinds of stemming algorithms: One is Lancaster stemmer, and the other is Porter stemmer. Two methods are quite similar, but it is known that the performance of Lancaster is better than the Porter.

Obs	Title(Original)	Title(Porter)	Title(Lancaster)
1	Marketing Intern	Market intern	market intern
2	Customer Service - Cloud Video Production	Custom service cloud video product	custom serv cloud video produc
3	Commissioning Machinery Assistant (CMA)	commiss machineri assist cma	commit machinery assist cma
4	Account Executive - Washington DC	account execut washington dc	account execut washington dc

[Table 2.1] The difference between Porter and Lancaster

After simplifying the data like above, we thought that the topic of text can be influential in detecting fake jobs. Therefore, we chose to do a LDA method.

LDA is the acronym of Latent Dirichlet Allocation. This is included in the Topic Modeling. Topic modeling is one of the statistical methods that help to discover the abstract topics in the text sets, and it is usually used in order to treat natural language. It usually help to find the hidden semantic structures which are hidden in the text paragraph.

The topic words appear frequently in the text, and using this, we can infer that the frequent words can refer to the topic of that text. Therefore, we can find out the topics. At that time, the degree of relation between topics and the real text can be shown mathematically, through

numbers.

LDA is one of these topic modeling algorithms. LDA assumes that the text is composed of the mixture of several topics, and topics create words based on the probability distribution. When the data is given, LDA tracks the text backwards so as to find out the process of text forming. Here is an example:

Sentence 1: I eat apple and banana.

Sentence 2: We love cute puppy

Sentence 3: My cute puppy is eating the banana.

With those simplified data, we started to do a LDA. At first, we used 'LatentDirichletAllocation' from sklearn in python. We set the number of components as 5, and used the batch learning method. Max iteration number is 25 After that, we used the 'title' variable to do the first LDA. The result are as below.

Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
Manag	servic	develop	english	design
engin	custom	sale	teacher	assis
analyst	associ	senior	abraod	market
softwar	web	busi	specialist	administr
account	develop	lead	director	engin
product	support	year	offic	system
project	time	intern	execut	sr
oper	repres	old	consult	digit
market	technic	repres	graduat	posit
junior	technician	apprenticeship	account	supervisor

[Table 2.2] Topic of Title

First topic shows the software engineer. Through the 'project', we can assume that those job titles include the word 'project' a lot. Also, through 'manag', 'market' or 'account', it seems that this topic also includes the management job postings related to marketing and accounting. Second topic shows the technical customer services, through developing web and supporting the services timely. Third topic shows the business & sales internship, and fourth topic shows English teacher and specialist who teach students in abroad. Last topic shows the supervisor or assistant in marketing & design part.

However, this method had a limitation: we need to pick out the words quite ineffectively, as I did above. Since we cannot do this for all text variables, we tried using another LDA package, which is called 'gensim'. It shows the weight for all topic words. For example, for the same title variable, the result is like below:

(0, '0.111*"engin" + 0.060*"design" + 0.046*"senior" + 0.045*"manag" + 0.038*"product"')

(1, '0.066*"manag" + 0.059*"market" + 0.047*"account" + 0.047*"assist" + 0.028*"offic"')

(2, '0.078*"english" + 0.077*"teacher" + 0.068*"abroad" + 0.043*"project" + 0.034*"year"')

(3, '0.145*"develop" + 0.066*"sale" + 0.033*"web" + 0.029*"director" + 0.027*"busi"')

(4, '0.073*"custom" + 0.073*"servic" + 0.048*"softwar" + 0.045*"associ" + 0.033*"engin"')

Using the above result, we made user-defined function that can make a table for the topic name that contains highest percentage, and the percentage of it, and the percentage of whole topics. Result of 'title' variables are like below, and we just picked first five rows to show the sample in this report.

Row number	Largest topic	Percentage of that topic	Each topics' percentage
0	Topic 1	0.4000	[(0, 0.06666717), (1, 0.40000263), (2, 0.06666...
1	Topic 4	0.5334	[(0, 0.36636603), (1, 0.03333552), (2, 0.03359...
2	Topic 4	0.6367	[(0, 0.04016087), (1, 0.2428372), (2, 0.040168...
3	Topic 1	0.4401	[(0, 0.040009614), (1, 0.44008118), (2, 0.0400...
4	Topic 0	0.7978	[(0, 0.7977691), (1, 0.050930295), (2, 0.05044...

[Table 2.3]

After that, to put this to the model, we rearranged the data into the matrix. Sum of rows are same as 1, which can be inferred that the each values is the percentage of containing each topics.

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Sum
0	0.066667	0.400003	0.066667	0.066667	0.399996	1.000000
1	0.366380	0.033336	0.033583	0.033336	0.533365	1.000000
2	0.040161	0.242837	0.040169	0.040158	0.636675	1.000000
3	0.040010	0.440074	0.040010	0.040271	0.439636	1.000000
4	0.797756	0.050943	0.050441	0.050432	0.050427	1.000000

[Table 2.4]

However, after proceeding the topics as 5, we found out that there are some limitations. Especially, we thought that the five number of topics is quite small to explain all of our data. Therefore, we changed the number of topics as 10.

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
0	0.033	0.700	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033
1	0.183	0.017	0.350	0.016	0.016	0.016	0.016	0.350	0.016	0.016
2	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.820	0.020
3	0.020	0.020	0.020	0.020	0.020	0.020	0.354	0.020	0.020	0.486
4	0.025	0.025	0.025	0.523	0.025	0.025	0.276	0.025	0.025	0.025

[Table 2.5]

4) Complex Data

First of all, Location data is separated by comma(',') and the sequence is (country code, state code, city name). Code follows ISO 3166 standard. Missing value rate was 1%, which is not that much. We put the 'missing' into those blanks. However, the problem of this column is that the information is quite partial. For example, for one row, it only has the country code like this –

(US, ,). Therefore, we thought this is unclear variable and cannot use this variable directly. Therefore, we separated it into three variables: country, state and city variable. After that, we filled out the missing values with 'missing'. Plus, using the country code, we added the continent column. These data would be used to finding the difference of fake job posting by country or continent.

	continent	country	state	city
1	America	US	NY	New York
2	Oceania	NZ	Missing	Auckland
3	America	US	IA	Wever
4	America	US	DC	Washington
5	America	US	FL	Fort Worth

[Table 2.6]

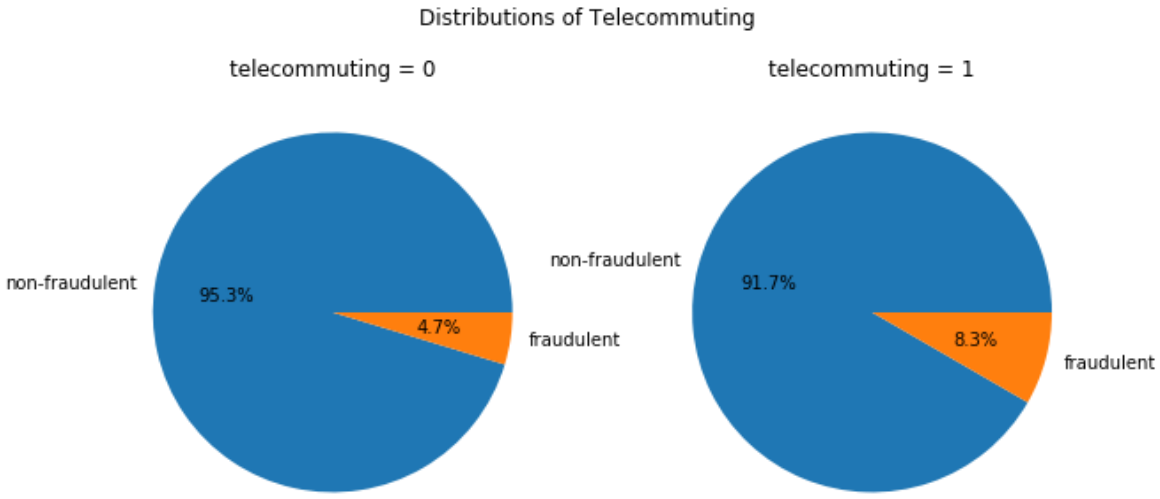
Also, in case of salary_range variable, we need to deeply look inside. Salary_range variable has very high missing value rate. It is about 84%, and written as "NaN". We substituted those missing values into the 0-0. This variable contains minimum and maximum wage of each companies, and it is the form of 'min – max'. After filling out the missing values, we tried to separate those values. However, there were some strange values: some of the values were totally not related to the salary. Therefore, we deleted those rows, which was about 28 rows. Plus, some of the companies ignored the unit (a dollar), and they just wrote in 10,000 dollar unit. Therefore, we changed those values after detecting, using for loop. After these process, we finally split the values into minimum and maximum salary. Plus, we thought that since there are a lot of missing values in this column, whether the company wrote the salary or not can be the meaningful information in detecting fake job posting. Therefore, we added the binary variable. In addition, there are two more variables that we made. One is the difference between maximum and minimum salary. The other is the ratio of difference and maximum salary.

3. Exploratory Data Analysis

EDA was conducted to determine whether the variables that have been pretreated can be used as significant variables in the real/fake classification. We first examined whether there are distribution differences according to the real and fake posts in the variables, and if the differences were significant, the variables would be judged as significant variables in the real/fake classification.

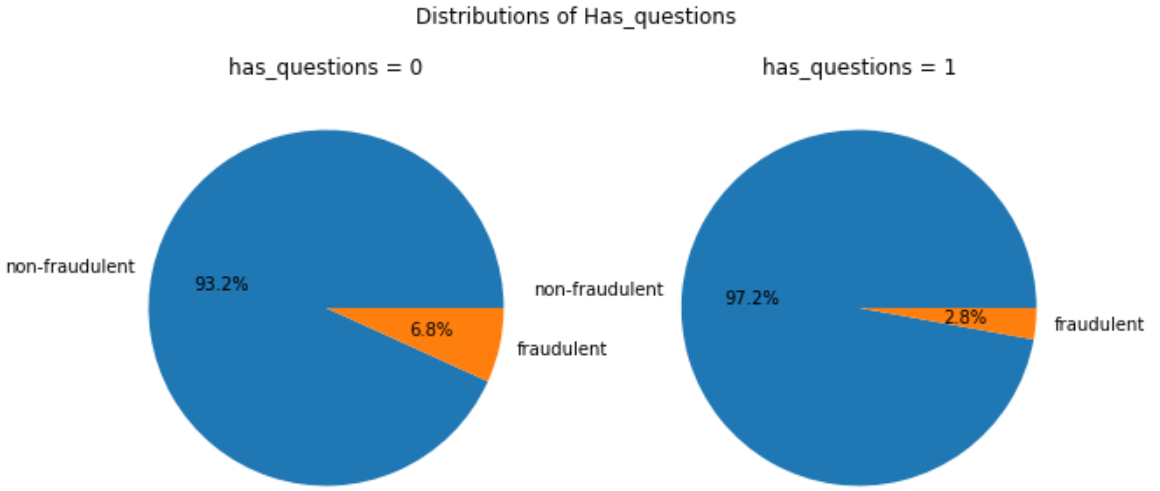
1) Binary Data

In the Binary data, each variable was divided by 0 and 1, looking at the ratio of fake and real post.



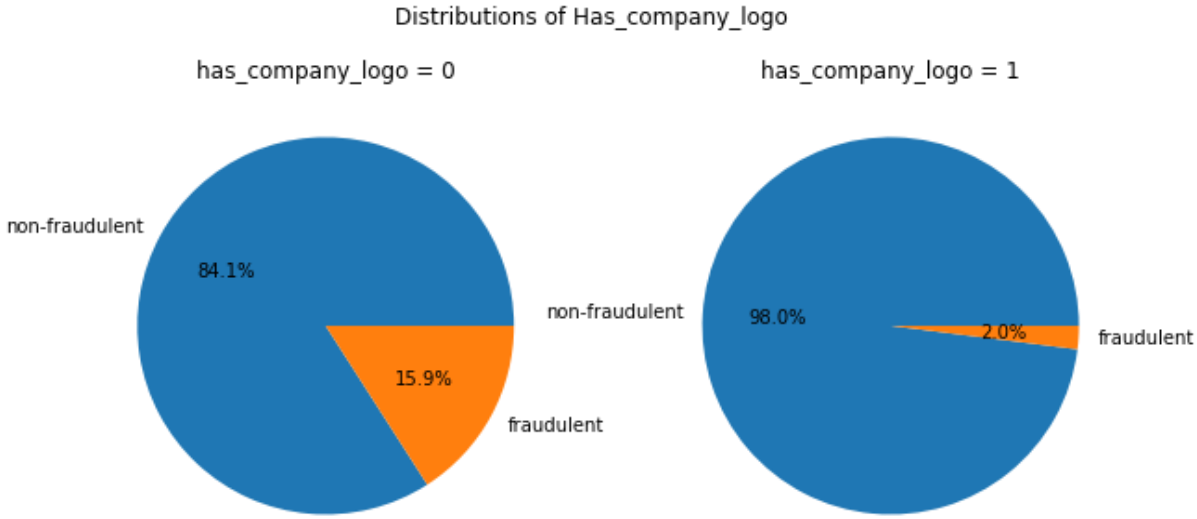
[figure 3.1] Distributions of Telecommuting

When Telecommuting is 0, the rate of fakes is 4.7% and when 1 is 8.3%, indicating that it is a little more likely to be fake posts when telecommuting exists as a condition. Other binary variables have a greater fake post ratio at zero, slightly different from Telecommuting.



[figure 3.2] Distributions of Has_questions

For Has_questions, the rate of fake posts at zero was 6.8% and 2.8% at one, especially for has_company_logo at zero at 15.9% and 2% at one, indicating that the difference is definitely greater than other variables.

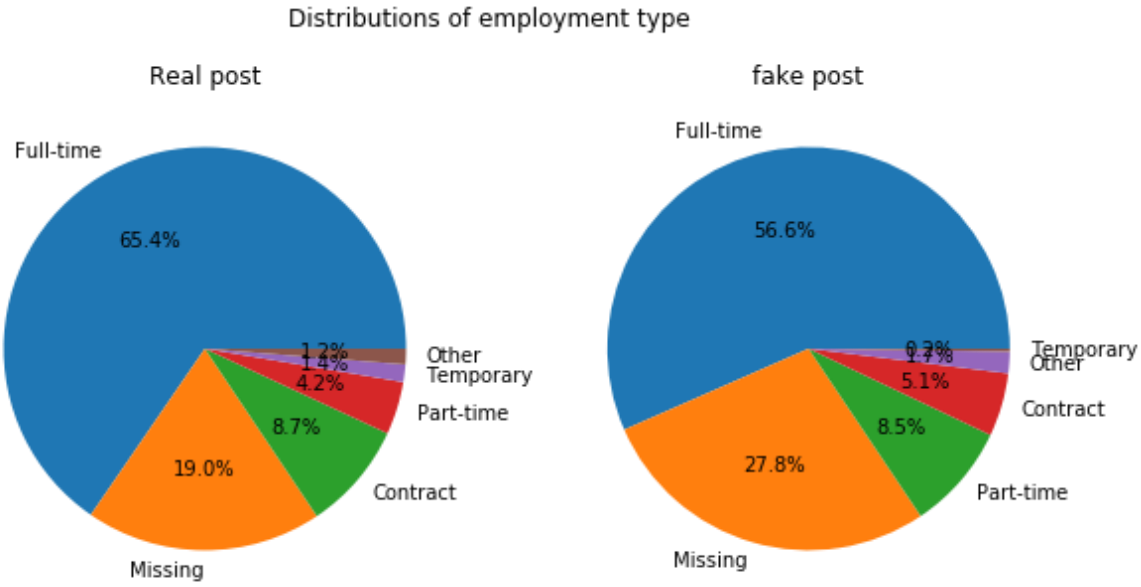


[figure 3.3] Distributions of Has_company_logo

As such, all three variables show some difference in distribution. To make this more accurate, we conducted a 2 sample t-test on $H_0 = no\ difference$ and all three variables were sufficiently small to reject H_0, which led us to conclude that there was a difference in the distribution of the variables.

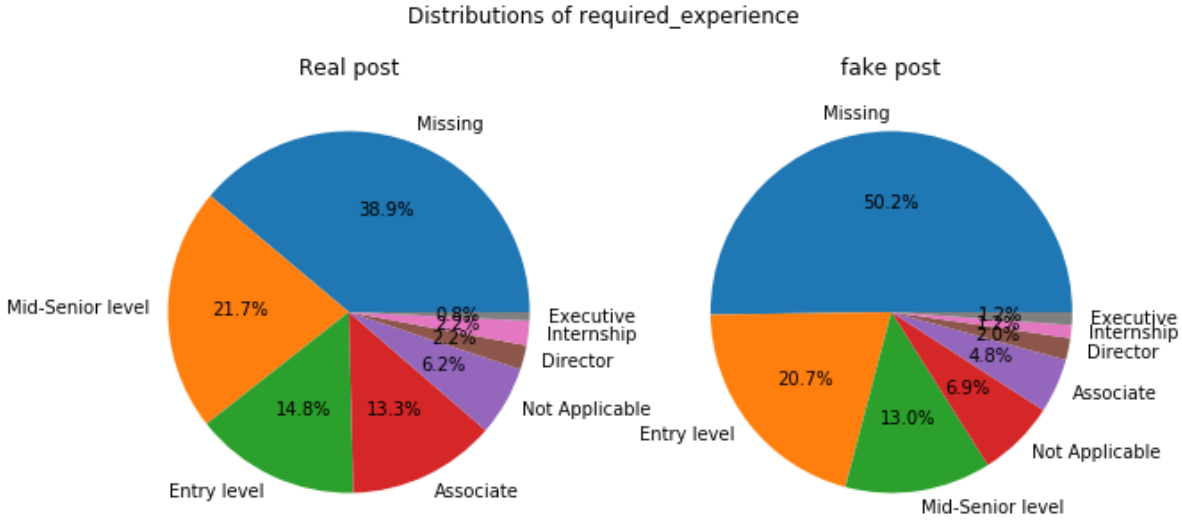
2) Categorical Data

As I mentioned in preprocessing part, three variables, 'departments', 'industry' and 'function', are grouped together and treated as text variables, and here we are going to talk about the other categorical variables except them. In the three variables 'Employment_type', 'required experience', and 'required_education', we divided them into 0 and 1 respectively, and then looked at the ratio of each category.



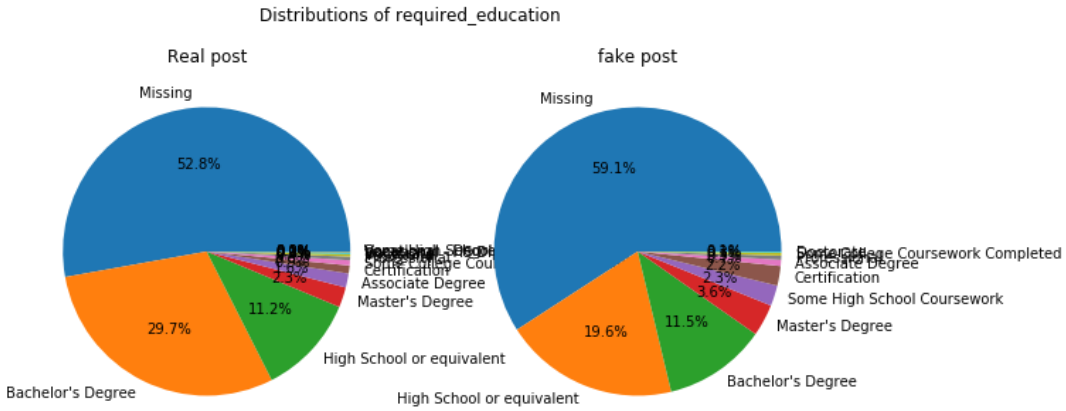
[figure 3.4] Distributions of employment_type

For Employment, missing value rate was about 9% higher and part-time was about 4% higher in real post than fake post. In other words, in the case of fake post, the Employment section was left blank or irregular workers were more likely to be hired than full-time employees.



[figure 3.5] Distributions of required_experience

In Required_Experience, too, the ratio of missing was 12% higher for fake post, and the ratio of entry level also rose. This shows that the fake post tend to require a lower level of experience than the real post.



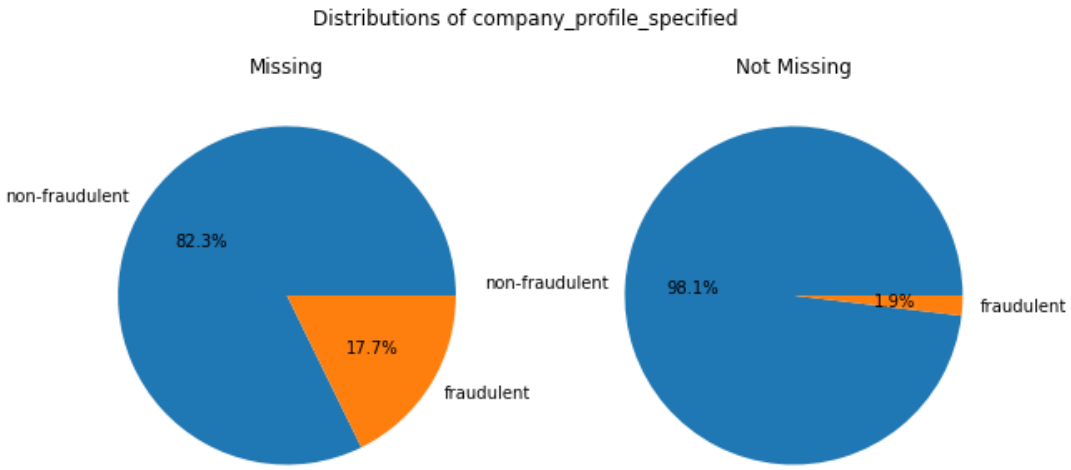
[figure 3.6] Distributions of required_education

This tendency is shown similarly in required_education, which also shows that the ratio of missing values in the fake post was high, and the proportion of high schools in the required academic background was high, indicating that relatively low academic performance was required.

As such, the demand for non-regular workers, low-level experience or academic background in the fake post was somewhat predictable, but we additionally checked the distribution comparison to make it more clear.

3) Text Data

In this text data, we wanted to apply three methods to a total of six text variables, including the addition of three variables from the Categorical variable. First of all, we checked whether missing could be a significant variable, then whether the number of words or the length of sentences could be a variable. Lastly, we selected the topic through LDA technique. Once we compared whether the difference between the real and fake posts is significant for each variable missing and the non-missing. As a result, the difference was particularly noticeable in the company_profile.

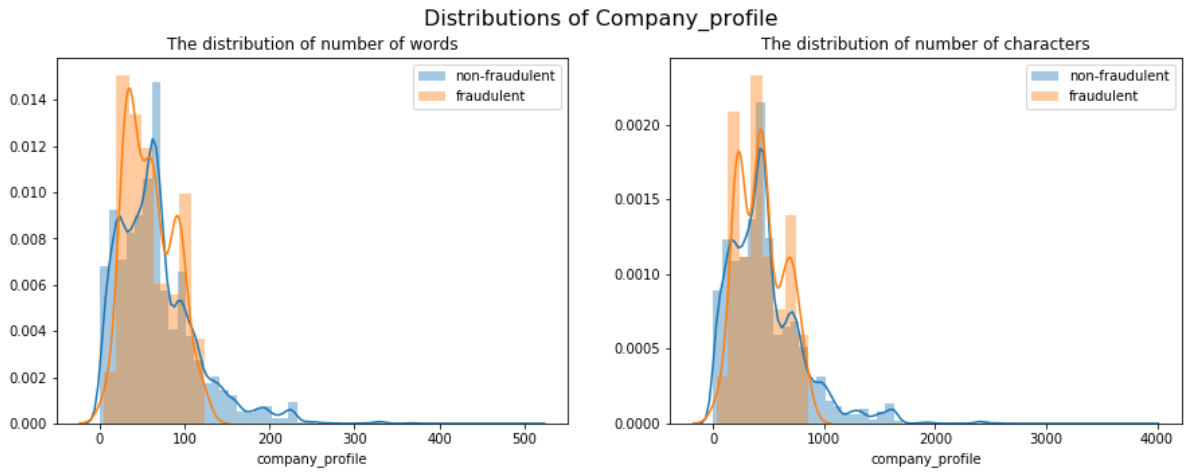


[figure 3.7] Distributions of company_profile_specified

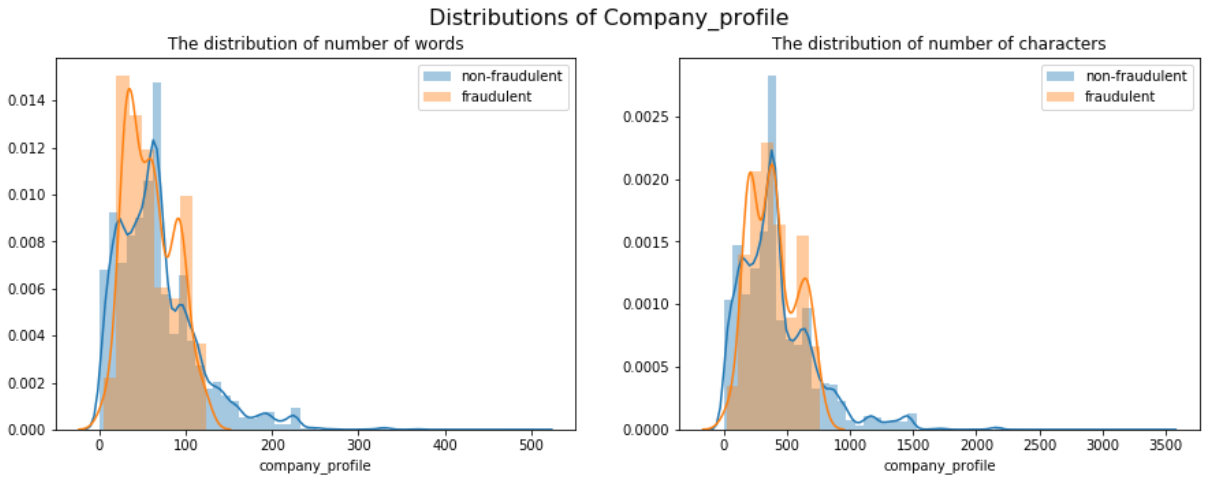
In the case of missing in company_profile, the ratio of fakes was 17.7%, about 16% higher than non-missing. In other words, it can be assumed that missing can be used as a variable to distinguish between real and fake in the text variable. T-test tests were also performed to much more accurately confirm this, although not visible to the naked eye, but the differences were also somewhat significant in ‘description’ and ‘requirements’.

Next, we wanted to determine whether the number of words or the length of the sentence could be a variable. However, there was a problem that even words with the same meaning could have various forms and meaningless parts, but to solve this problem, the stemming algorithm was applied. The stemming algorithm removes the tangent, etc. from the modified word and separates the stem of the word. Two of the most famous stemming algorithm, Porter and Lancaster, were applied to this analysis.

We looked at the distribution of the number of words and the length of sentences in the data processed by the algorithm, but neither Porter nor Lancaster found much difference in the distribution except for the company_profile variable.

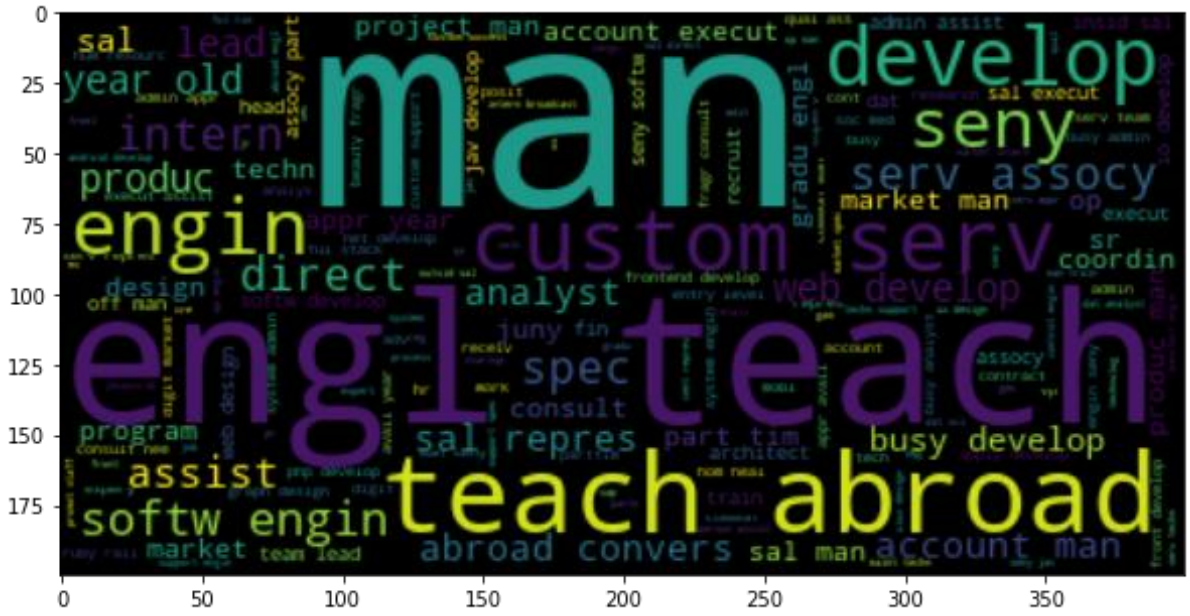


[figure 3.8] Distributions of company_profile_porter

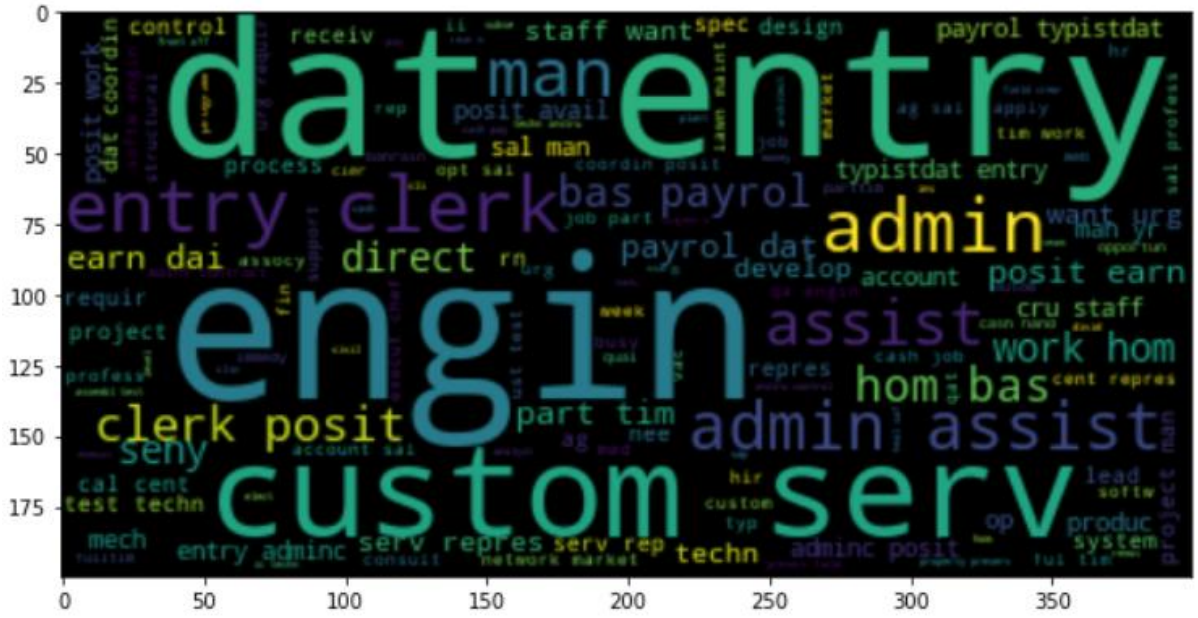


[figure 3.9] Distributions of company_profile_lancaster

We have also used word cloud techniques to see how the text variables themselves are organized, not the length of the sentences or the number of words. As a result, there was a difference in the lowest value in the title processed using the lancaster technique. The results of this visualization are as follows:



[figure 3.10] Word cloud of title (real post)



[figure 3.11] Word cloud of title (fake post)

In addition, we thought about the derived variables that can be made from text variables and tried to implement them through LDA. This method extracted 10 topics from each variable and subsequently obtained the weight each topic occupies in the data. A total of 60 additional topic variables were created for six variables. There were slight differences in the distribution of fake/real post for each variable. Therefore, we decided that these variables were also available for the project.

4) Complex Data

First of all, in the case of location, rather than looking at countries with many categories, we divided them into five continents and looked at whether there was a difference in the proportion of real/fake in them. Again, to see if the difference is significant, the t-test test for $H_0 = no\ difference$ was carried out, and the results were as follows.

	Africa	America	Asia	Europe	Oceania
p-value	0.09013	3.432e-8	0.092	2.2e-16	0.03019

[table 3.1] p-values for each continent

The above results show that there are significant differences in America, Europe, and Oceania. More specifically, the rate of fake was significantly lower in Europe, and the rate of fake was significantly higher in the Americas and Oceania. This made it possible to use it as a variable, knowing that there was a difference between continents.

Next, we looked at whether missing can be a significant variable and what the distribution of salary maximum and minimum is, respectively, using the new salary-related variables previously created for the salary_range. It was expected that the rate of fake post would be high in the data without wages, but on the contrary, the rate of fake post in the data with wages marked was

higher. To determine whether these differences were significant, a t-test was conducted and it was possible to conclude that the p-value differed by 0.000. Next, we looked at the distribution for maximum and minimum salary. As a result, the values of maximum salary and minimum salary in the fake post were generally low, indicating that relatively low wages were written on the fake post.

4. Special Method: Oversampling

1) What is Oversampling?

It is a kind of copying the data. We can randomly copy, or we can set the copying standard. Usually, we use oversampling when the class of data is imbalanced. For example, assume that we need to sort out faulty products among whole products. Since the proportion of anormal data is extremely low, it is really difficult to grasp the trend and predict the result. Although the accuracy is high, recall decreases rapidly, as there are few anormal data. Therefore, people use the Oversampling method. There are two representative oversampling methods: One is random oversampling, and the other is SMOTE. In Random oversampling, we simply choose rows from the minority class randomly, and add it into the minority class directly. This usually results in overfitting problem. To prevent overfitting problem, SMOTE(Synthetic Minority Oversampling Technique) came out. First of all, decide the needed number of minority class. After that, It finds out the k-nearest-neighbors(k is hyperparameter) of each minority points. Connect the KNNs with the original point. After that, generate new points, which can be located in a half position of the line, or a quarter/ three quarter of the line. These values are resulted from the surrounding data, so SMOTE can reduce the probability of overfitting.

To use SMOTE, there are some precautions. First, you should do SMOTE only for training data, not for whole data. If not, artificial data can be problematic in test process. Second, you should tune the hyperparameter. Especially, when we do SMOTE, we should decide the ratio of copying. For example, if we choose 1 as a ratio, it means that the number of majority and minority class become same. The standard of tuning can be the F1 score, and we can find out the hyperparameter that can result in highest F1 score.

2) Why do we need Oversampling?

Our data is highly imbalanced data. Target variable is binary type, compounded of 0 and 1. 0 means that the job posting is real one, and 1 means that the job posting is fake. Fake rate of this variable is only 4.4%. Therefore, even if the predict model regard all data as 0(real), the accuracy rate becomes 0.95, and this result is actually not that accurate. Therefore, we decided to use oversampling method to overcome this problem.

3) Problems that we faced during doing oversampling

First problem that we faced was about the text and the categorical variables. Since they were all string types, oversampling could not be adjusted directly. Therefore, we need to vectorize all values. In case of categorical variables, since the kinds of categories were limited, we could easily make it as sparce data. However, since text variables have a lot of words, the number of columns increased enormously.

We processed the data through one-hot coding anyway, but some problems were occurred. First, the data itself. We chose SMOTE to prevent overfitting problem. In case of SMOTE, it uses K-nearest neighbors and make new data sample. However, it does not consider the distance with the data of other class, which can cause the overlapping of class. Also, it is not suitable for high-dimensional data(Han et al, 2020). Therefore, with our high-dimensional data, it was not that adequate to use SMOTE.

We tried to use SMOTE with our data, but the memory that is needed to do SMOTE was insufficient. Also, the result was not good, since the model using oversampled data resulted in algorithm which classify data just with target. Therefore, we decided to stop using this data, and use some models that we learned during courses for predict and interpretation.

5. Modeling Methods & Result

In addition to the variables added above, much more variables were used in models than the original data. At first, there were three classification techniques: LASSO, Random forest, and XGboost. However, all three techniques were used solely for accuracy, so we decided to apply additional Logistic regression for interpretation. A total of four models were used in this project.

1) LASSO, Random forest, XGboost

A little bit of explanation for each model is as follows. Once LASSO is a model in which the model prevents overfitting by adding constraints on linear regression coefficients. Add a constraint to minimize the sum of the absolute values of the weights, limiting the size of the unnecessary coefficients. In other words, the advantage is that the coefficient of an unnecessary variable is zeroed and only the necessary variables can be used.

Next, I will explain about random forest. In order to understand Random Forest, you need to know the Decision Tree technique, which is to find the rules of the data and classify the data by repeating the binary classification, such as Yes/No. The random forest repeats these decision trees several times, which can be extracted from the existing data by random sampling to repeat the decision tree and to find more accurate rules based on the repeated decision tree.

XGboost also repeats the decision tree several times, such as the random forest, but unlike the random forest, which was used to extract data by random sampling, XGboost picks samples and then adjust weights for the next sample. This method is called boost, and XGboost is a technique that prevents overfitting and adjusts it faster than other boost. As such, the accuracy model will use three models to prevent overfitting.

As described earlier, we tried to apply some to the problem of data imbalance, but this failed to deal with the text-type variable, requiring all variables to be converted back to numeric. Numerical variables attempted to convert through standardscaler, categorical variables one-hot encoding, and text variables Tfidfvectorizer. However, at this time, the text variable was sparse matrix and the other variable was just matrix, which caused an error because the type did not fit.

To solve this problem, a new function was created, but memory was not able to handle it, and eventually modelling was carried out by applying some after adjusting the number of words used in Tfidfvectorizer.

When LASSO, random forest, and XGboost were applied, there was a problem in which the overall accuracy would increase if all data were expected to be zero due to unbalanced data. So, for more detailed verification, we wanted to look at all three of the predicted values of rate1= p(real fake value l predicted fake value), rate2= p(predicted fake value l real fake value), and the rate=total accuracy. The ‘correct rate’ consists of [rate, rate1, rate2]. The hyper-parameters were adjusted to the most appropriate 'correct rate' to find the most suitable model. Also, cut off was changed based on the found hyper-parameter. The cut off was adjusted from 0.4 to 0.6 because it was determined that using too extreme a value would result in logical errors.

In the case of LASSO, when alpha = 0.09 and cut off = 0.5, the highest accuracy was corrected rate = [0.6456, 0.0451, 0.444], so we decided to adopt this value. For random forest, we chose the correct rate = [0.942, 0.0952, 0.0741] when cutoff = 0.4, n_esti=100. In the case of nodes, more than 100 had similar results, but the number of nodes became too large to adopt. Finally, XGboost had the highest cored rate when the number of nodes was 70, and cut off seemed to be very irrelevant, so we used 0.5 which is commonly used. The correct rate is [0.9341, 0.0741, 0.0741]. The results for each model can be found in the following table.

Actual	0	1
Predicted		
0.0	478	15
1.0	254	12

[Table5.1] LASSO

Actual	0	1
Predicted		
0	713	25
1	19	2

[Table5.2] Random forest

Actual	0	1
Predicted		
0	707	25
1	25	2

[Table5.3] XGboost

The results of the classification were not very satisfactory. Especially when looking at Random Forest and XGboost, the overall accuracy is high, while the accuracy for fakes is not very high. In order to solve this data imbalance problem, we had to try oversampling in various scenarios, but we had to go with default only because there were many problems. In addition, there were limits to the lack of memory due to the increased dimensions of the data, which resulted in the inability to vary the number of max_features and the oversampling. Unfortunately, these problems have resulted in poor accuracy.

2) Logistic regression

Logistic regression is a regression analysis and a model used for binary classification. The logit transformation reduces the range of the predictors between 0 and 1, and the predicted values indicate log odds. Once you know the log odds, you can also calculate P(y=1) as well. This model

can be interpreted from the perspective of odds, which shows how much the odds of success increase as variables increase. In this project, we applied a logistic regression model when the fraudulent was 1, i.e. fake post.

The number of columns of data is too large, and all of these data forms are inconsistent, making it difficult to fit the entire data. Eventually, the entire data was divided into four parts. It was divided into Topics variables/Text Feature (the number of sentences and words in the Text variables)/calary related variables/category variables, each suitable for logistic. The text section did not use the original data, but instead two data obtained by preprocessing with the porter and Lancaster algorithms were used.

Topics is a subset of data that consists of numbers in which words belonging to topics obtained from LDA appear in observations of text variables. More precisely, it consisted of the probability of calculating the proportion of 10 topics for each document. Because the topic was chosen at random through the LDA, the value of all variables may be zero for documents with a large proportion of topics other than the 10 topics.

Sixty topical variables were extracted from a total of six types of text variables, and the data obtained from Porter and lancaster were also fit separately. Significant coefficients were found in both company_profile for data processed by Porter and company_profile and requirement for data processed by lancaster. In particular, both algorithms showed that the regression coefficient for company_profile_topic was very large, indicating that it was the variable that had the greatest impact on the probability of fakes. For a more detailed explanation, I would like to give an example of company_profile_topic0 of the porter data. The regression coefficient for this variable is 89.68, and if the proportion of topic0 in the company_profile increases by 0.1 the odds at P(y=fake) increase by $e^{89.68}$times. At this time, the units of the topic are 0 to 1. In lancaster, the regression coefficient of the requirement is significant, but its value is negative. In this case, the greater the proportion of topics, the less likely it is to be faked. The results vary slightly depending on which algorithm is used, but because the coefficients for company_profile_topic are close to 90 for the porter and 50 for lancaster, it can be concluded that the higher the number of topics in the company_profile, the higher the probability that the ad is faked.

In the Text Feature section, the number of words and sentences were appropriate. However, the number of words and sentences were not independent, so they became suitable separately. The results showed that the number of words and sentences in both Porter and Lancaster were significant variables. However, when looking at the coefficients, the effect on the odds of fakes, which is mostly close to zero, was very minimal. Thus, it was determined that while the variables in Text Feature are related to the probability that they are fake posts, the extent of the impact is very weak and negligible.

Salary-related variables include salary_max, salary_min, salary_differentiation, which is the difference between max and min, and salary_ratio, which finally divided the differences into max. Because these variables are also related variables, it is not good to fit all of them. Therefore, they fit each of the four. As a result, the p-value was significantly lower at 2e-16 only in the salary_ratio. The regression coefficient for salary_ratio is 1.63, and the greater the value of the

ratio, the greater the odds of fakes being $e^{1.63}$ times and the greater the probability of fakes being faked. The greater the difference between Maximum and minimum, the greater the probability of faking exists.

Finally, we proceeded with the suitability of the category variables. There are four variables and each variable has significant dummy variables as follows: Employment type = [Full-type(coefficient:-0.23), Contract(-0.69), Temporary(-1.67)], Required_Experience = [Mid-senior level(-0.38), Associate(-1.04), Entry level(-0.3)] Required_eduction = [Bachelor's Degree(-0.75), Master's Degree(0.77), High school(0.45), Certification(0.94),high school coursework(3.67)], Continent = [America(1.54), Oceania(1.71)]. The baseline of the previous three variables is missing and the baseline of continent is Africa. For both the Employment type and Required_Experience, the probability of fakes is lower when there are more significant dummy variables when the condition is blank because the regression coefficient is negative. For Required_eduction, the coefficient for Bachelor's Degree is negative and the coefficient for the remaining values is positive. This means that the probability of faking is lower when Bachelor's Degree is conditioned, but the probability of faking increases if other degrees, too high or too low, are written. Finally, for Continent, the probability that the regression coefficients are both positive numbers in America, Oceania, and fake post increases.

In the logistic model, we looked at how the probability of a fake post would change depending on the variable. We knew that dual company_profile_topic had a significant impact on this probability, and also that there were some significant variables. we think this activity is meaningful in that these meaningful variables were not directly derived from the data, but were created by applying various transformations and techniques.

6. Result

1) Achievements

First, we preprocessed all variables well. Especially, in case of text variables, we tried to understand in three steps: at first, we removed the stopwords. After that, we picked stemming words only, and lastly, we used LDA method. Using these data, we made some new variables and draw graphs to grasp out some meaningful result. Also, in case of complex data, we split into several variables to use. Also, through logistic regression model, we could grasp out the relative importance of each variables.

Also, when we started the analysis, we wondered if the variables would really play a big role in determining fake job openings. However, data analysis has shown the influence of some factors. In particular, we could see that not only the conditions of job advertisements but also the influence of company-related information was somewhat great. Usually, when people see job ads, they focus on conditions such as salaries, not on company-related information. However, the study also showed that we should look closely at the company information in the job ads that we ignored.

2) What we need to do more

First, we could not solve the imbalanced data problem. It was quite hard for us to treat the text

variable in oversampling. We need to study more about imbalanced data soothing methods that can also be used in text variables. We could not solve this problem, so the modeling result was not that great.

Second, we thought that using other modeling methods could improve the quality of results. For instance, we can additionally apply SVM method. Also, some deep learning methods can be the better alternatives. It's because this data, especially the preprocessed data for modeling, have a lot of variables, deep learning method could improve the result, although we cannot understand the inner structure of analysis.

References

1. Data Reference

Data : [Real or Fake] Fake JobPosting Prediction, https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction

ISO Country Code: ISO Country Codes – Global, https://www.kaggle.com/andradaolteanu/iso-country-codes-global

2. Papers

한정수, 이채현, 고경찬, 우종수, 홍원기(2020). GAN을 활용한 비트코인 불법거래 과 표본 추출 기법, 2020 통신망운용관리 학술대회

정현승, 강창완, 김규곤(2008). 불균형 데이터에 대한 오버샘플링 효과 연구, 한국자료분석학회, 10권 4호, 2089-2098.