上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Machine Learning Final Project

Topic	:	Recruitment Scam Detection
Group	:	Group 8
Teacher	:	李成璋老师
Group Member	:	张泽宇
		Amanda Jane Kusno
		Bryan Zhen Dong Aw
		Junhee Park

Recruitment Scam Detection

Introduction

In the old days before the internet and personal computers became part of our daily life, people used to find jobs through newspapers, yellow pages and other platforms. Some of them even had to knock on many doors to find jobs, limited by their access to job-seeking channels. After the industrial revolution and technological revolution, the demand for both skilled and unskilled workers started to rise, followed by the number of mediums that can be used by employers or companies for hiring workers. Until today, tons of websites, applications, and social media platforms are specifically designed or used for recruiting labor force. Although modern technologies have brought us the convenience of finding or choosing jobs, they have also become a new medium for scammers. People who were scammed by believing online recruitment advertisements are now everywhere to be seen, some crucial cases ended up with financial problems. Therefore, our group came up with an idea that we hope could analyze the features of recruitment advertisements and determine which of them are the characteristics of scams by implementing algorithms learnt from the Machine Learning Class.

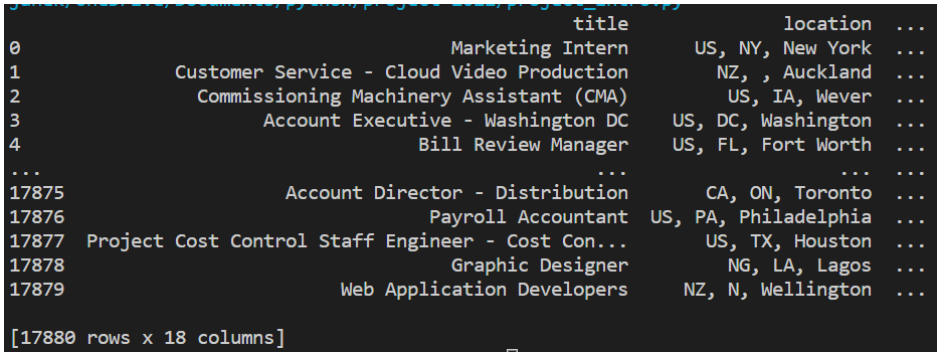
Data and Models

For the data source, we had been using data collected by The Employment Scam Aegean Dataset (EMSCAD), which contains 17,880 real-life job ads published between 2012-2014. From the collected data, we planned to predict which of the variables (title, location, department, company profile, job description, requirements, benefits, telecommunicating, company logo, employment type, experience, industry, employment type, etc) have the biggest correlation or key determination on identifying fraud job ads (fraudulent). By using different parameters, models, hyperparameters, etc. for every variable, we divided the test into several trials to help in identifying the accuracy of the model to achieve the best outcome for analyzing the prediction. In this project, we had used models such as Multi-Layer Perceptron, NLP(Natural Language Processing) network,

Logistic Regression, Support Vector Machine, and an ensemble learning of the models mentioned above. We then conclude which model is best/better by using information such as accuracy and true/false positive rate.

Data Preprocessing

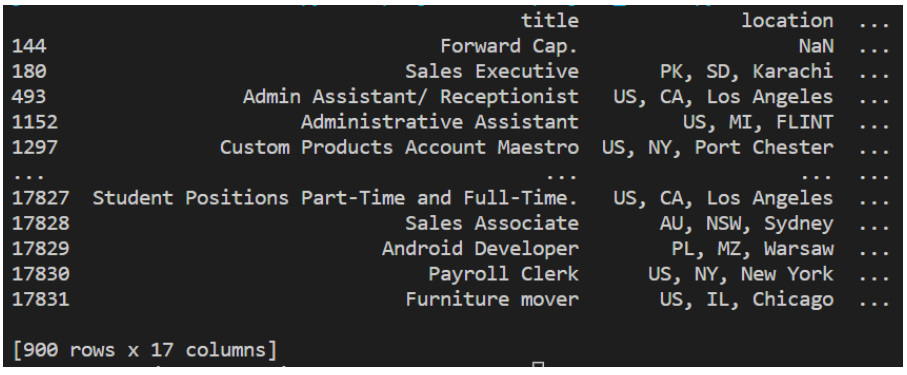
The data preprocessing is divided to 4 steps. The first step is reducing. From the data that we get, we first need to choose the data in general that is needed to be trained, and based on the reference that we find, we'll choose the data which is in imbalanced result. From this process, the data from around 18 thousand will be reduced to 900.



	title	location	...
0	Marketing Intern	US, NY, New York	...
1	Customer Service - Cloud Video Production	NZ, , Auckland	...
2	Commissioning Machinery Assistant (CMA)	US, IA, Wever	...
3	Account Executive - Washington DC	US, DC, Washington	...
4	Bill Review Manager	US, FL, Fort Worth	...
...	...	...	...
17875	Account Director - Distribution	CA, ON, Toronto	...
17876	Payroll Accountant	US, PA, Philadelphia	...
17877	Project Cost Control Staff Engineer - Cost Con...	US, TX, Houston	...
17878	Graphic Designer	NG, LA, Lagos	...
17879	Web Application Developers	NZ, N, Wellington	...

[17880 rows x 18 columns]

Figure 1. Data before reducing, 17800 data



	title	location	...
144	Forward Cap.	NaN	...
180	Sales Executive	PK, SD, Karachi	...
493	Admin Assistant/ Receptionist	US, CA, Los Angeles	...
1152	Administrative Assistant	US, MI, FLINT	...
1297	Custom Products Account Maestro	US, NY, Port Chester	...
...	...	...	...
17827	Student Positions Part-Time and Full-Time.	US, CA, Los Angeles	...
17828	Sales Associate	AU, NSW, Sydney	...
17829	Android Developer	PL, MZ, Warsaw	...
17830	Payroll Clerk	US, NY, New York	...
17831	Furniture mover	US, IL, Chicago	...

[900 rows x 17 columns]

Figure 1. Data after reducing, 900 data

Next, considering the ‘location’ which includes country, state and city, we first separate them from the column ‘location’ and create three new columns, which are ‘country’, ‘state’ and ‘city’, then remove the feature ‘location’.

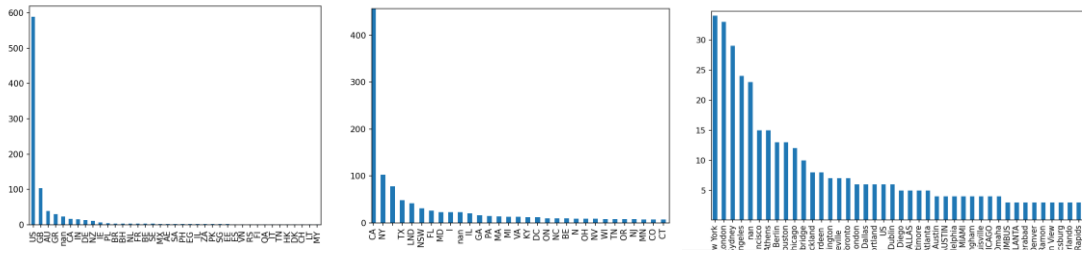


Figure 3,4,5. Visualization after separating ‘location’ to ‘country’, ‘state’ and ‘city’. The ‘US’, ‘CA’, ‘New York’ is the most value in each training dataset.

The second step is cleansing. The data we get also includes columns like description, benefits, requirements which contain a lot of explanation like the one we used to see on ads, and it also includes URL, Email, phone which consist of a bunch of random alphabet or number. Simply the step is like the one attached here, we input the data and use regular expression to replace it. On the overall explanation, we use ‘HTML Parser’ and ‘String Io’ to help us transform it to a more visualize model.

Experience and Skills Required

- Minimum of 3 years of sales experience
- Ability to work in a home office environment
- Exceptional speaking, writing, and negotiation skills
- You must be a motivated self starter and instil that others
- College degree

Experience and Skills Required
- Minimum of 3 years of sales management experience
- Ability to work in a home office environment
- Exceptional speaking, writing, and negotiation skills
- You must be a motivated self starter and instil that in others
- Experience in hiring and training individuals
- College degree

Figure 6,7. Before and after using HTML parser and String IO

Third step is transformation. We had also extracted the characters(char) that were used in the ‘title’ in case that we are using them to train our model. In the same time, we calculated the percentage of caps lock for each title and put it as a new feature ‘title_caps’.

Next, the number of bullets, paragraphs, and bold words in the features (description, company profile, requirements and benefits) were calculated and the common logarithm of the numbers above were squeezed into the range of 0-2. Other than that, we also replace the text of URL, phone and email to make them simpler to avoid complexity in data. The number of URLs in an ad might be useful in predicting whether the ad is a fraud or not, so we also considered the number of URLs in our dataset.

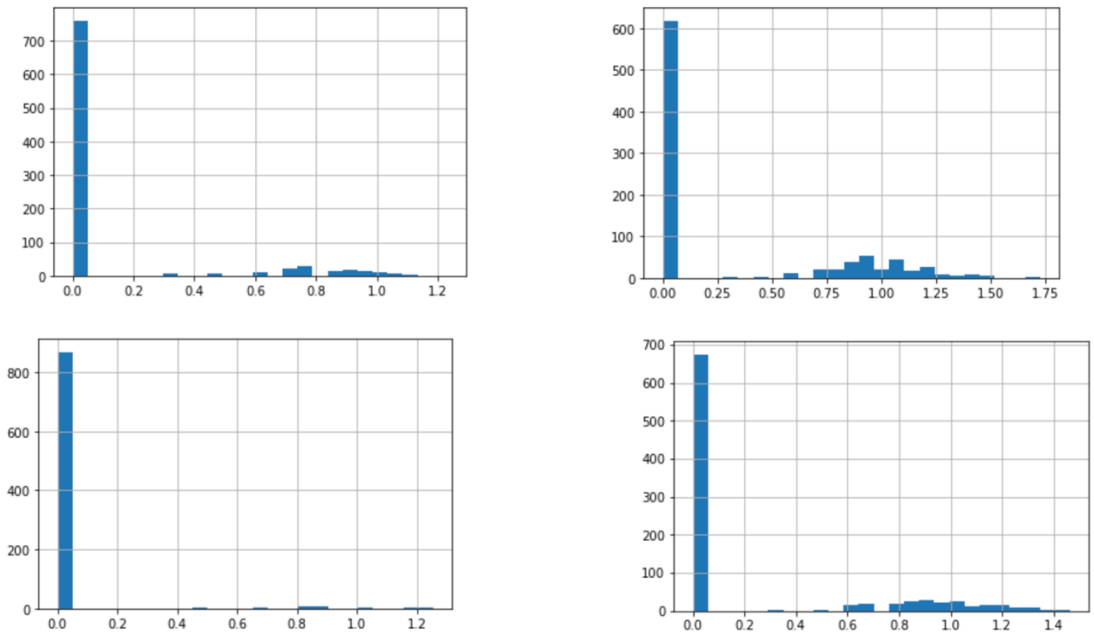


Figure 8,9,10,11. The normalized result of counting bullets for each features (Description, Company profile, Requirements, Benefits)

Then the data pre-processing is finished. There are other kinds of variables in this data. The dummy variables such as ‘telecommunicating’, ‘has company logo’, ‘has questions’ are the binary features which consist True or False, whereas ‘fraudulent’ is our dependent feature in this model. Also, there are other features such as ‘Employment type’, ‘Required experience’, ‘Required education’, ‘Function’, and ‘Industry’.

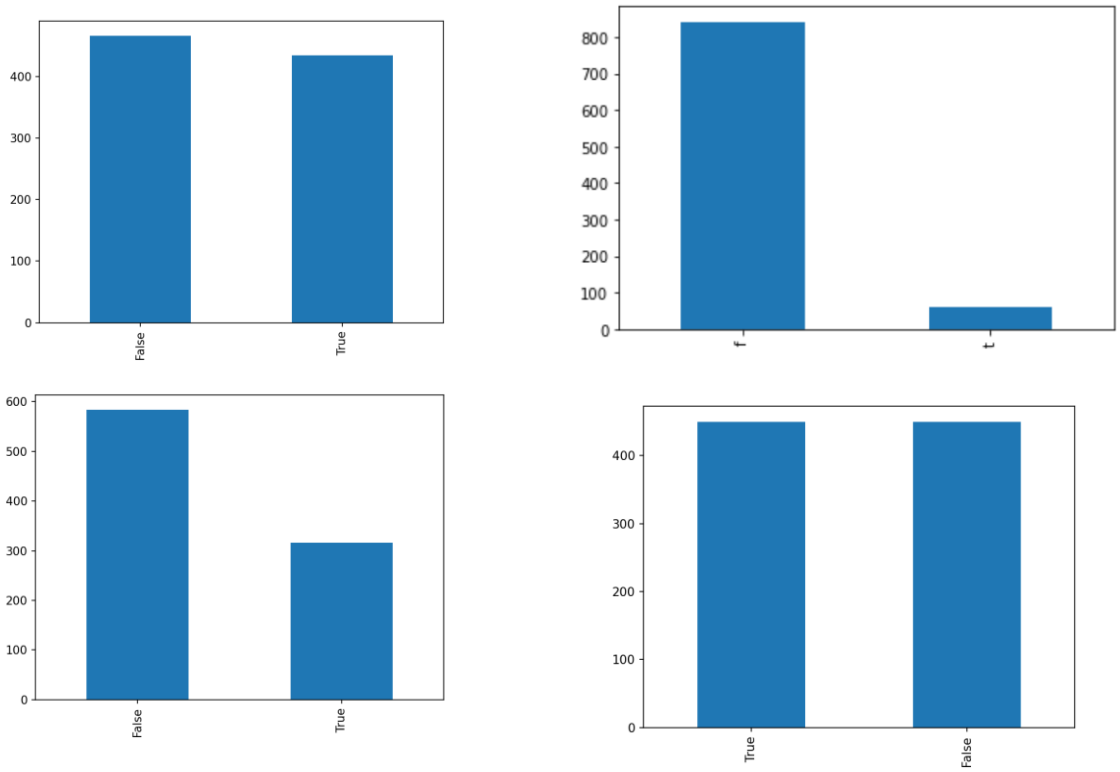


Figure 12, 13, 14, 15. 4 binary output variables (has company logo, telecommunicating, has questions, fraudulent)

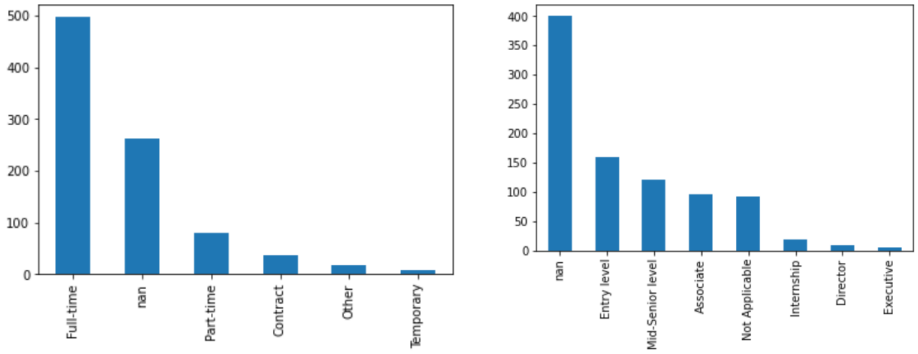


Figure 16, 17. Visualization of ‘Employment type’ and ‘Required experience’

Using Multi-Layer Perceptron model

After the preprocessing and additional feature extraction process described above, we are able to use a MLP (Multi-layer Perceptron) model to utilize these features and make a prediction. We used characteristics of text (number of bullets, paragraphs, bold, URLs), country, employment type, dummy variables (telecommunicating, company logo, questions), required experience, industry as an input variable to construct this MLP model. The MLP model takes an input of (5,

80), and then used Reshape layer to flatten that to a 1-D vector, applicable to Multi-Layer Perceptron. The MLP model is consisted of 2 hidden layers and outputs a 1/0 value as prediction. The accuracy of the MLP model is 0.883.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 400)               0

 dense (Dense)               (None, 10)                4010

 dense_1 (Dense)             (None, 8)                 88

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 4,107
Trainable params: 4,107
Non-trainable params: 0
_________________________________________________________________
```

Figure 18. Summary of the MLP model. It consists of input layer, reshape layer, hidden layers each consists of 10, 8 nodes and ‘adam’ activation function, output layer with ‘sigmoid’ activation function.

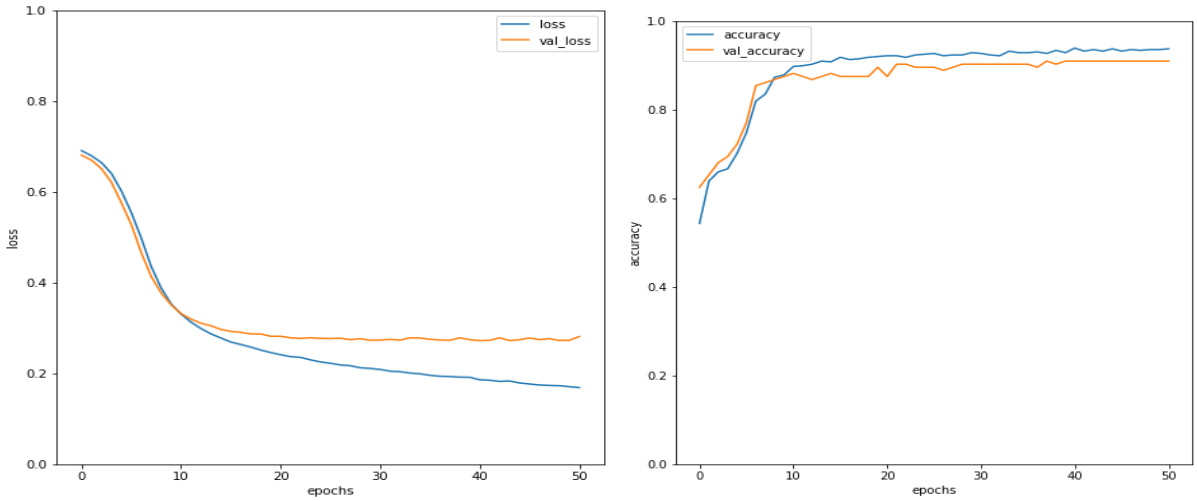


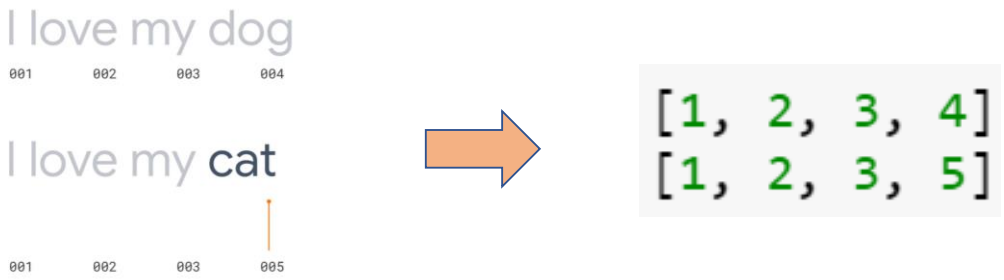
Figure 19, 20. The visualization of loss and accuracy on MLP model. The validation accuracy converges to 0.883

Natural Language Processing

Next, we used 4 text fields of title, description, benefits and requirements to train the NLP (Natural Language Processing) model. The difference between Multilayer perceptron and natural language processing is that the multilayer perceptron uses numerical and categorial input data while this NLP model is used to deal with natural language data. In our dataset, we have four

features that is in the form of text, which is title, description, requirements, and benefits. These four features went through the process of tokenization and padding, then they were used to train 4 NLP models respectively.

Tokenization is the process of turning different words into numbers. When all the words were assigned to specific number, we then use padding to form the sentences or paragraphs of the input data. In this process, we used keras.Tokenizer to convert the text to integer vectors, and then used padding to unify their length.



Explanation 1. The brief explanation of tokenization process. In this process, all words are assigned to specific integers.

```
array([[356, 357,   0, ...,   0,   0,   0],
       [  6,  53,   0, ...,   0,   0,   0],
       [ 18,   5,  46, ...,   0,   0,   0],
       ...,
       [ 95,  12,   0, ...,   0,   0,   0],
       [  9,  10,   0, ...,   0,   0,   0],
       [334, 857,   0, ...,   0,   0,   0]], dtype=int32)
```

```
array([[  4, 405, 128, ...,   0,   0,   0],
       [ 40, 368,   0, ...,   0,   0,   0],
       [  5,   1, 609, ...,   0,   0,   0],
       ...,
       [  1,   1,   1, ...,   0,   0,   0],
       [ 42,   1,  13, ...,   0,   0,   0],
       [272, 734,   3, ...,   0,   0,   0]], dtype=int32)
```

Figure 21, 22. The result after tokenizing and padding to title and description.

Using an embedding layer as input, the tokenized data then sent to a NLP network for prediction. For each of our NLP networks, we have 1 average pooling layer, 1 hidden layer with 24 nodes and ‘relu’ activation function followed by the output layer with ‘sigmoid’ activation function. Same as the previous MLP model, we also implemented binary cross entropy method as the loss function and ‘adam’ as the optimizer in the networks. The accuracy of NLP models are **1)** 0.833 taken title data as input, **2)** 0.872 taken description data as input, **3)** 0.778 taken requirements data as input , **4)** 0.767 taken benefit data as input.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding (Embedding)        (None, 17, 16)            16000

 global_average_pooling1d (G  (None, 16)               0
 lobalAveragePooling1D)

 dense_3 (Dense)              (None, 24)                408

 dense_4 (Dense)              (None, 1)                 25

=================================================================
Total params: 16,433
Trainable params: 16,433
Non-trainable params: 0
_________________________________________________________________
```

Figure 23. Summary of the NLP network using 'title' as an input

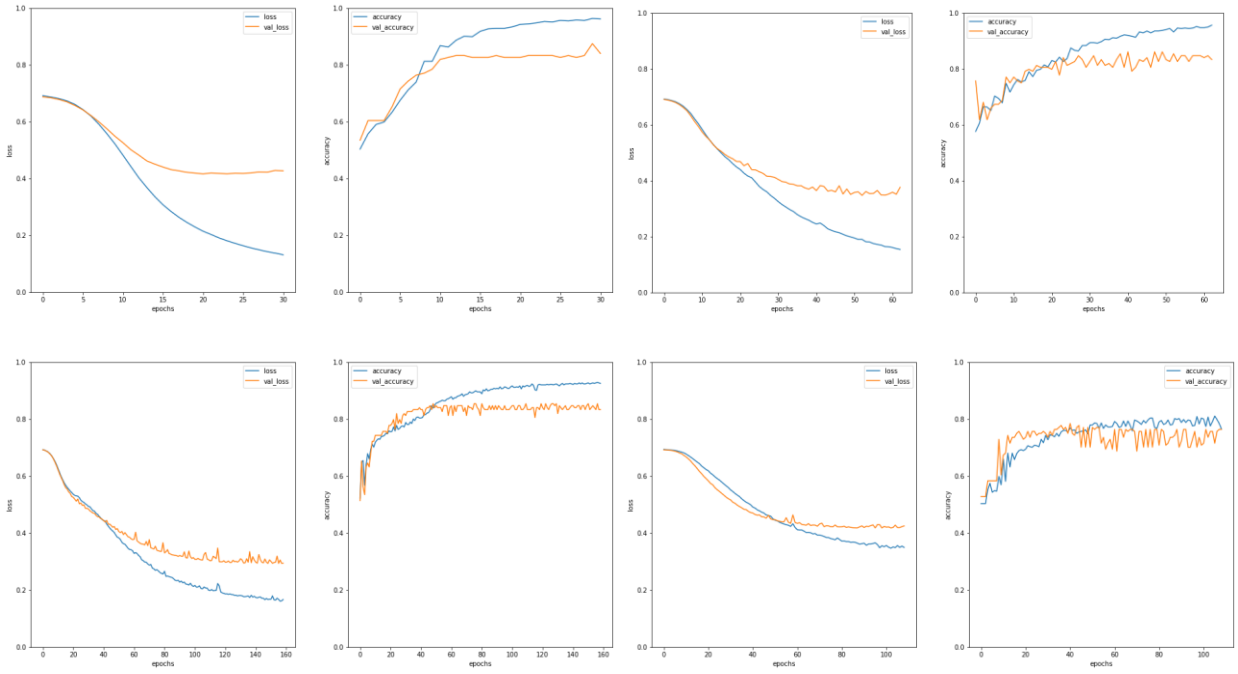


Figure 24, 25, 26 ,27. The graph of loss and accuracy for each feature (title, descriptions, requirements, benefits). The network with description as input data achieve the highest accuracy on test set, which is 0.872.

Other Classifiers

Neither the accuracy of the Multilayer Perceptron nor the Natural Language Processing networks could satisfy us, so we came up with another two different classifiers which are

logistics regression and support vector machine using the tokenized data to test if we could find other algorithms which perform better.

At first, we implement logistic regression on tokenized data. These are the evaluations on the 4 Logistics Regression model using different text feature. We can clearly see that they all performed better than a model with random guessing. The accuracy of logistic regression models are **1)** 0.6 taken title data as input, **2)** 0.656 taken description data as input, **3)** 0.694 taken requirements data as input, **4)** 0.644 taken benefit data as input. Among all the 4 text features, model that is trained with 'requirements' has the best performance with an accuracy of 0.694 and 0.702 for the area under the curve.

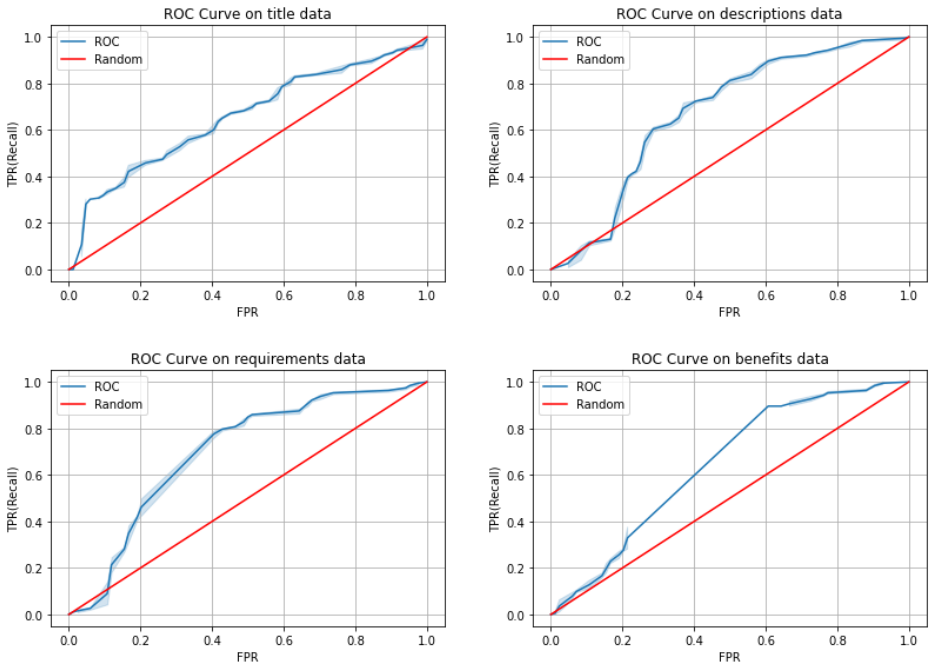


Figure 28, 29, 30, 31. The ROC curves of logistic regression model (title, descriptions, requirements, benefits)

The next is the support vector machine (SVM). Same as the previous one, SVM also works better than random guessing. The accuracy of each SVM models is **1)** 0.716 taken title data as input, **2)** 0.7 taken description data as input, **3)** 0.7 taken requirements data as input, **4)** 0.7 taken benefit data as input. In the case of SVM model, it performed better with the text feature 'title', which has the accuracy of 0.716 and the highest area under the curve, 0.796. When we were training SVM models, we use grid search cross-validation method for finding best penalty coefficient and gamma which perform the best.

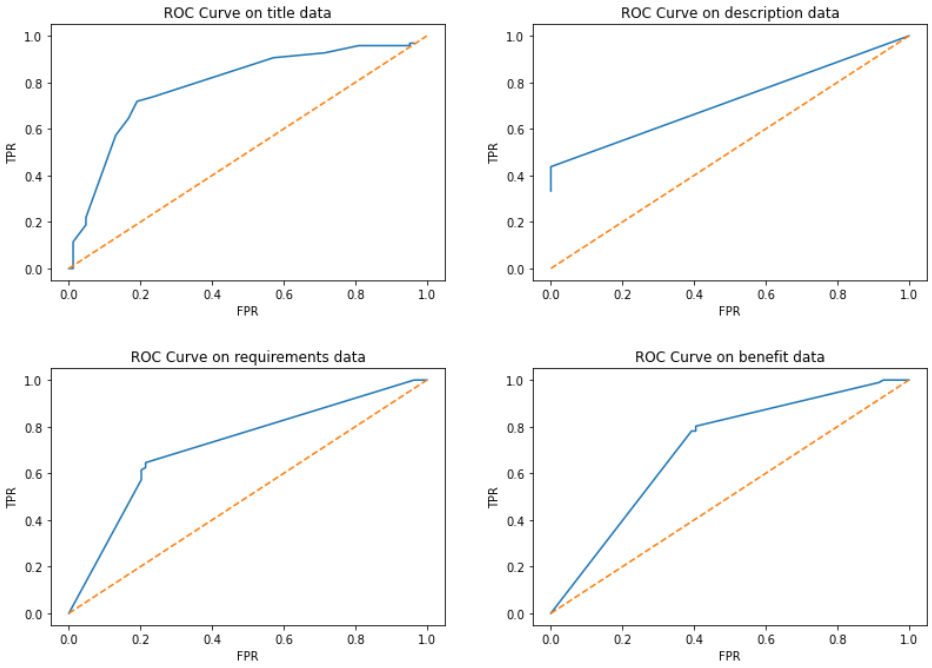
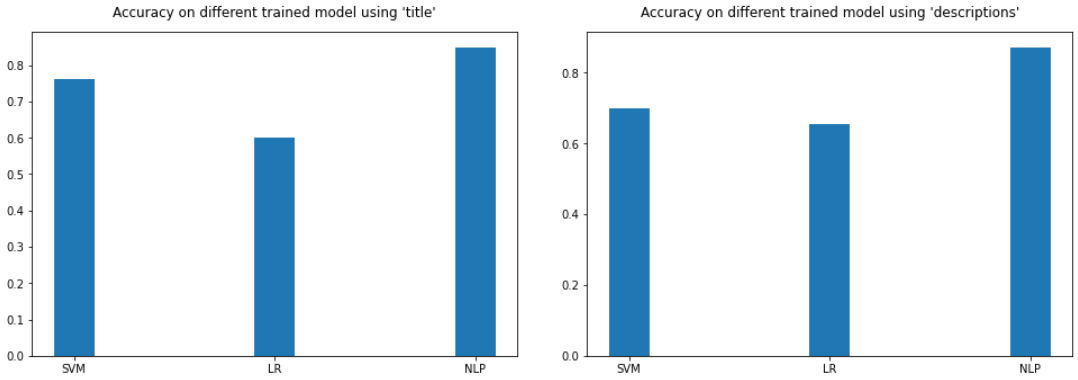


Figure 32, 33, 34, 35. The ROC curves of logistic regression model (title, descriptions, requirements, benefits)

We can conclude that the NLP networks has the highest accuracy no matter what the feature text is used to train the network. Since we had calculated the area under the curve for both logistics regression and SVM, we can conclude that it is better for us to use SVM rather than logistics regression because the roc curve also indicates the performance of a classifier.



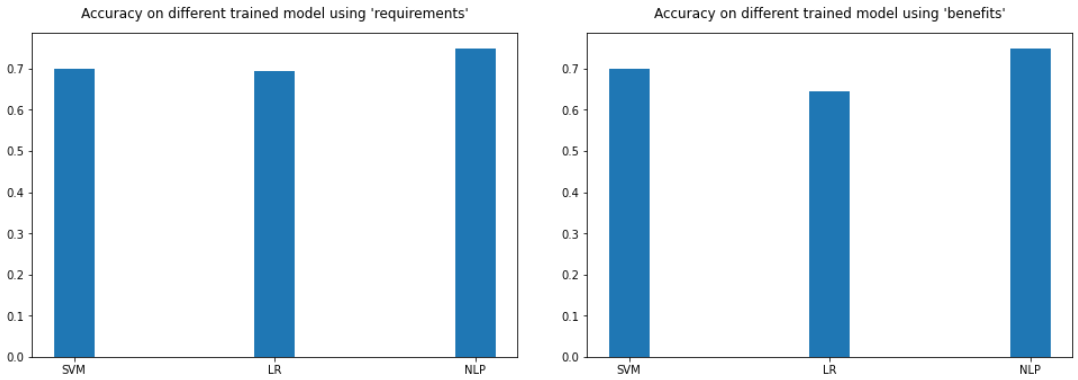


Figure 36, 37, 38, 39. The comparison among models. As we can see, NLP neural network model get a highest accuracy above all of features.

Ensemble Learning

Those individual models still cannot satisfy us. Therefore, we came up with the ensemble learning, thinking that it might be useful to combine the predictions of the models. For the methods of ensemble learning, we have ‘another’ neural network and voting in our project.

For another neural network, we used the output (prediction score) from the 4 Natural Language Processing networks and the multilayer perceptron model to train a new neural network. Just like this, and the criterion of the network is same as previous, which included 1 hidden layer with ‘relu’ activation function and the output layer with ‘sigmoid’ activation function, followed by binary cross entropy loss function and ‘adam’ optimizer. The achieved accuracy is 0.872.

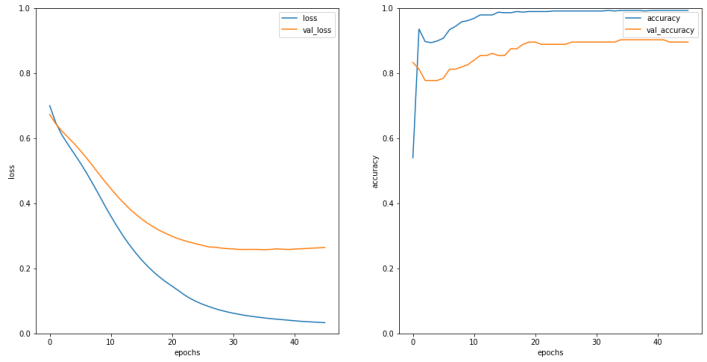


Figure 40, 41. The loss and the accuracy of combined neutral network

Other than that, we also excluded the multilayer perceptron by just combining the output of 4 NLP networks to train another neural network to make a comparison. For the result, we can see that both neural networks have the same performance, and the accuracy is the same with the NLP model which uses ‘description’ as the training feature. In comparison, this kind of ensemble learning does not help much in increasing the accuracy of prediction and it also uses longer time to train as we first need to train several neural networks and combine their output to train another one. The other problem of this ensemble learning method is that it seems to overfit the training set.

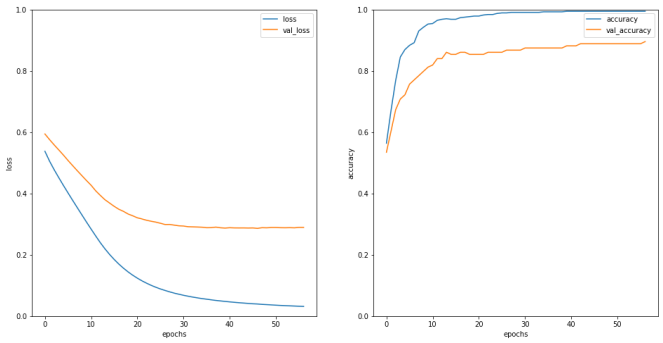
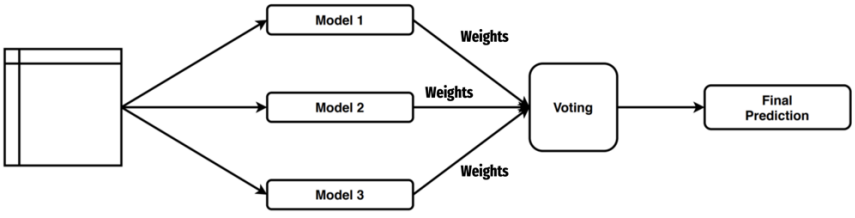


Figure 42, 43. The loss and the accuracy of combined neutral network excluded MLP. As we can see, the result has no significant difference.

The next ensemble learner is the voting algorithm. We first extract the prediction of each model and times them by their weights and sum them up to get a new prediction. In this case, we had taken the test accuracy of each model to act as the weights. The prediction of each model is given positive 1 or negative 1 based on whether the recruitment ad is a fraud or not. The new prediction is based on the sign of the summation, if it is positive, the recruitment ad is considered a fraud, and vice versa.



$$H(x) = sign\left(\sum_{i=1}^{n} h_i(x) \times w_i\right), \qquad h_i(x) = \begin{cases} +1, & if\ fraudulent \\ -1, & if\ not\ fraudulent \end{cases}$$

$$w_i = test\ accuracy\ of\ model\ i$$

Explanation 2. The algorithm of voting system.

We had evaluated 4 voting systems, which are the neural networks of 4 NLP and 1 multilayer perceptron and the one without multilayer perceptron. We also have included the logistics regression and SVM models in this voting algorithm. For the evaluation part, we had used the 10-fold cross validation method and compute the average accuracy and standard deviation for each type of voting. The accuracy of each voting system is **1)** 0.92 taken 4 NLP and 1 multilayer perceptron, **2)** 0.944 taken only 4 NLP, **3)** 0.84 taken logistic regression, and **4)** 0.942 taken SVM. We can see that the accuracies are all increased than its original model. In this case, voting without the multilayer perceptron works the best as it achieved the highest accuracy and smallest standard deviation among the others, and the second goes to the SVM. It is bit strange that the voting without the MLP model performs better than the one that included it, so we came up with the inference that the comprehensive dataset seems to be a distractor in this voting system.

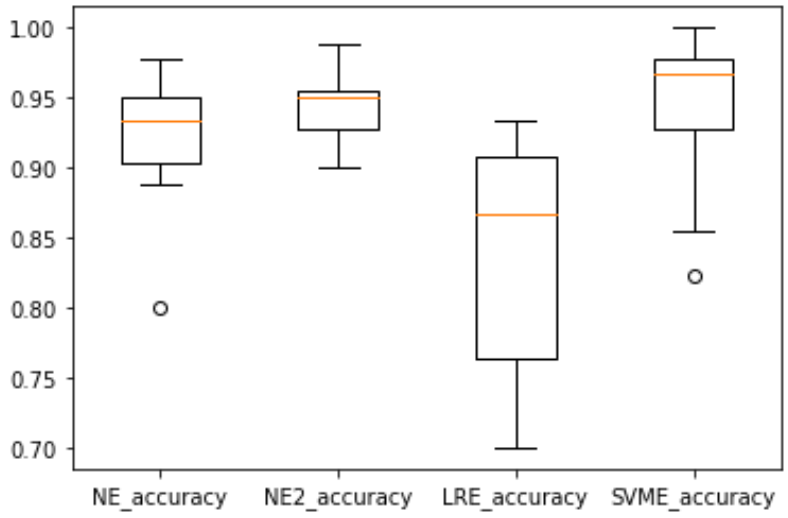


Figure 44. Boxplot of the result of voting algorithm. (4 NLP+1MLP, 4 NLP, logistic regression, SVM)

Conclusion

To make the best recruitment scam detection classifier, we first preprocessed dataset to make it easy to handle. In data preprocessing, we also made some variables that are remarkable when dealing with our problem. We first implemented multi-layer perceptron model with all of letter features. Then, to utilize letter features, we tokenized and padded each feature, trained several neural networks and generated predictions with the models. Also, we use classic classifiers like logistic regression and SVM too. To achieve higher accuracy for prediction, we combined 4 NLP networks and MLP model, and just combined 4 NLP networks. Lastly, we used voting algorithm for each model, then we could get a higher accuracy than before.

Voting seems to work well with algorithms like neural networks and support vector machine, but this method should not be treated as a one-size-fits-all as it might perform badly on other algorithms like the logistics regression. On the other hand, this kind of ensemble learning could help us in achieving a shorter training time but higher accuracy as we don't need to train a single model with all the information and have them as input data. Despite the advantage of voting, we need to be cautious that some input data might become distortions and lead to a drawback in the voting algorithm. At last, due to the limitations of current knowledge, our group might omit some other ways or algorithms that could perform better, for example we could not include the "comprehensive dataset" in our logistics regression and SVM models due to the dimension of this array.