**Data Mining and Predictive Analytics**

**(BUDT758T)**

**Project Title:** ​**Real or Fake job posting**

**Team Members:**

Jiashu Wang, Ziyi Li, Yijie Fan, Liangrui Lu,Yingnan Mu

**ORIGINAL WORK STATEMENT**

**We the undersigned certify that the actual composition of this proposal was done by us and is original work.**

|  |  |  |
| --- | --- | --- |
|  | **Typed Name** | **Signature** |
| Contact Author | Liangrui Lu |  |
|  | Yijie Fan |  |
|  | Ziyi Li |  |
|  | Jiashu Wang |  |
|  | Yingnan Mu |  |

**TABLE OF CONTENTS**

**EXECUTIVE SUMMARY 1**

**DATA DESCRIPTION 1**

**DATA PREPARATION AND EXPLORATION 3**

**RESEARCH QUESTIONS 4**

**METHODOLOGY 5**

**RESULTS AND FINDING 7**

Naive Bayes 7

Random Forest 8

Association Rule 9

K-Nearest-Neighbor 11

Classification Tree 12

Model comparison: Accuracy, Sensitivity and Specificity 12

Model comparison: ROC Curve and Lift Charts 12

Bi-LSTM 12

**CONCLUSION 15**

**REFERENCE 16**

**APPENDIX 16**

1. **EXECUTIVE SUMMARY**

What is the most important factor in a company? Nowadays, we believe manpower is the key in the future. Company who has the best crow will lead the future in their society. Company provided various compensation plans to attract future Canadians. People have the ability to choose what they want. Our team picked this topic because we think it is very valuable for a company. We will determine whether a job posting is real or not. From exploratory data analysis, we were encouraged to implement various machine learning models and compare their results. This professional dataset we found from Kaggle.

# II. DATA DESCRIPTION

The dataset was prepared by This dataset contains 18K job descriptions out of which about 800 are fake. The data consists of both textual information and meta-information about the jobs. The dataset can be used to create classification models which can learn the job descriptions which are fraudulent.



Dataset: <https://www.kaggle.com/code/vikassingh1996/fake-job-post-prediction-countvec-glove-bert/notebook>, with 18000 rows of job posting information, and 18 columns.

**Train\_Test\_Split**: test\_size = 0.25, random\_state=16, saved as:

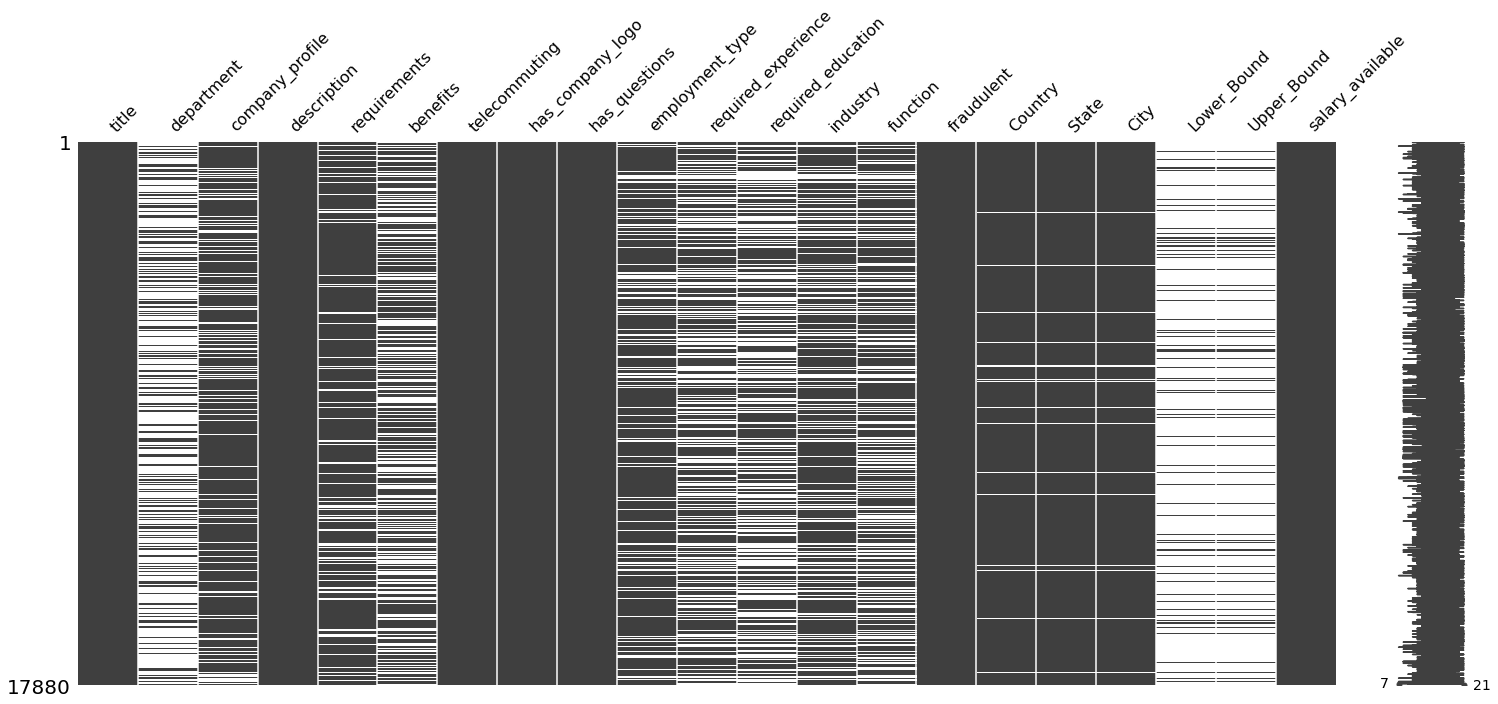
Train Set: Real\_Fake\_Prediction\_Cleaned\_Train\_2.csv (use validation split in it, if needed)

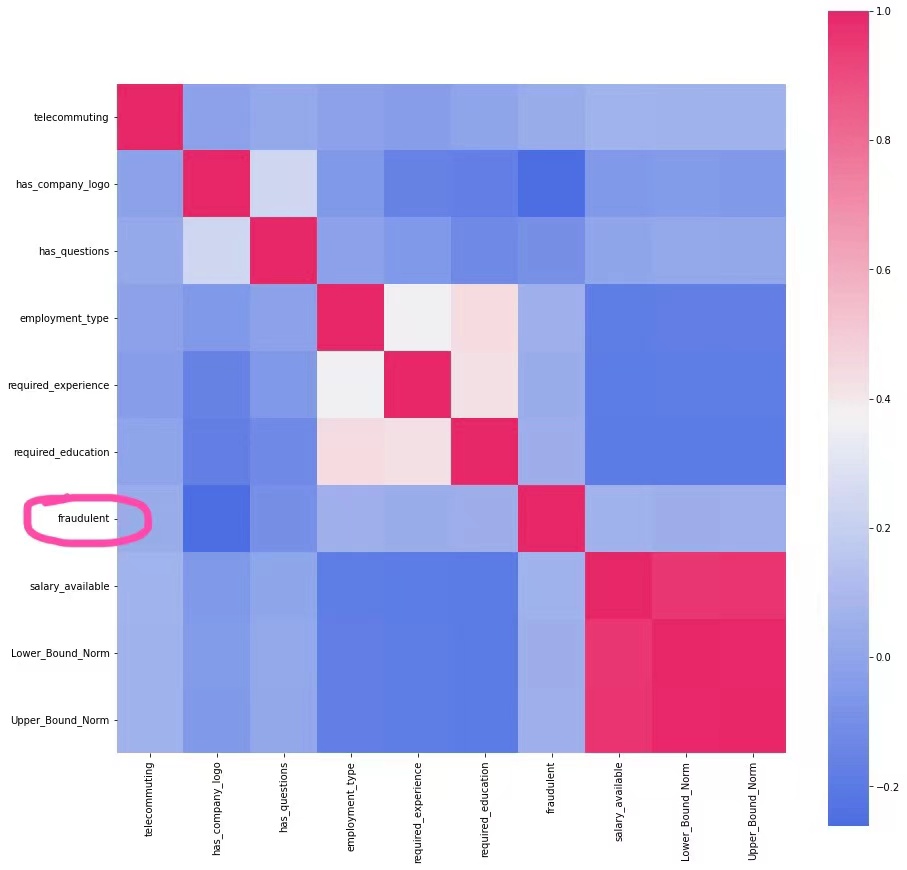
Test Set: Real\_Fake\_Prediction\_Cleaned\_Test\_2.csv (used ONLY for testing)

Columns: (black for text columns, blue for categorical columns, red for numerical columns)

* Job\_Id：Unique job ID.
* Title: The title of the job ad entry.
* location: Geographical location of the job ad.
* Department: Corporate department.
* Salary\_range: Indicative salary range
  + Lower\_Bound
  + Upper\_Bound
  + salary\_available
* Company\_profile: A brief company description.
* Description: The details description of the job ad.
* Requirements: Enlisted requirements for the job opening.
* Benefits: Enlisted offered benefits by the employer.
* Telecommuting: True for telecommuting positions.
* has\_company\_logo: True if company logo is present.
* has\_questions: True if screening questions are present.
* Employeement\_type: Full-type, Part-type, Contract,etc.
* Required\_experence: Executive, Entry level, Intern, etc.
* Required\_education: Doctorate, Master’s Degree, Bachelor, etc.
* Industry: Automotive, IT, Health care, Real estate, etc.
* Function: Consulting, Engineering, Research, Sales etc.
* Fraudulent: target - Classification attribute.

# III. DATA EXPLORATION AND PREPARATION



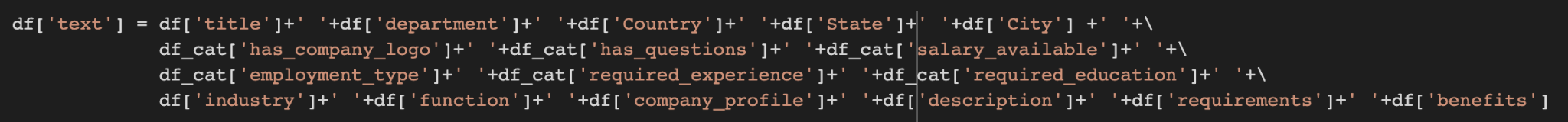


Feature of data:

* From the first graph, it’s clear that our dataset contains lots of missing data, which should also be regarded as factors, since fake jobs tend to leave more columns blank.
* Most of the columns are given in text format, and dummies are given as 0 and 1.
* From the heatmap, we are able to claim that there are no clear relationship between *fraudulent* and any numerical or categorical data; nevertheless, some categorical variables, like *has\_company\_logo*, are related to *fraudulent*

# Data Preprocessing:

1. Fill missing values as a long self-made word, like: nodepartment, noprofile, changing it to a meaningful token for future text mining.
2. Change column *salary\_range* to 2 numerical columns: *Lower\_Bound*, *Upper\_Bound*, then create categorical variable *salary\_available*, 1 for available and 0 for not.
3. Change column location to 3 columns: *Country*, *State*, *City*.
4. Numerical variables Normalization: *Lower\_Bound*, *Upper\_Bound,* by filling Null with 0.1, 0 with 1, then applying log function (original value 0 will be 0, original Null will be -1), then applying normalization using: , to normalize value range to [0,1]
5. Similarly to 1., mapping dummies to a long self-made word, like: havecompanylogo
6. Using LabelEncoder from sklearn to encode categorical data to indexes, in order to get heatmap for also categorical variables.
7. Concatenate all texts and self-made word to a text with all messages:



# IV. RESEARCH QUESTION

Real or fake job classification is a challenging natural language processing (NLP) problem, with a board application that could also benefit ourselves as job seekers. Our team wanted to explore working with unstructured data and analyze the effect of machine learning algorithms on job classification.

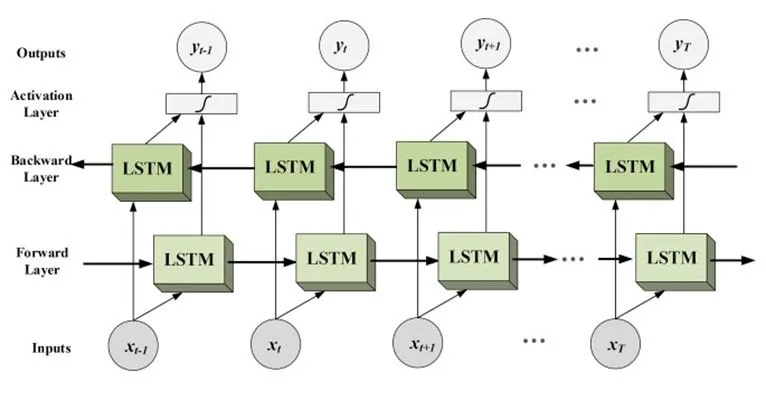
We choose this dataset to illustrate the real or fake job posting will affect the readers to find out the right solutions that truly determine their future pace. We could detect if the job posting is real or fake using a machine learning method. This effect not only will have a huge impact with the Canadians themselves, but also help the company to find out their best match. It will help all different industries, not limited to the business field.

During our project, we will use different models to compare the best accuracy as well as provide reasonable recommendations. Besides that, we also combine different technical tools, such as python, R, tableau in order to give readers complex perspectives as well as secure abundant information. This research will help the job posting more appropriately and develop the society in the future.

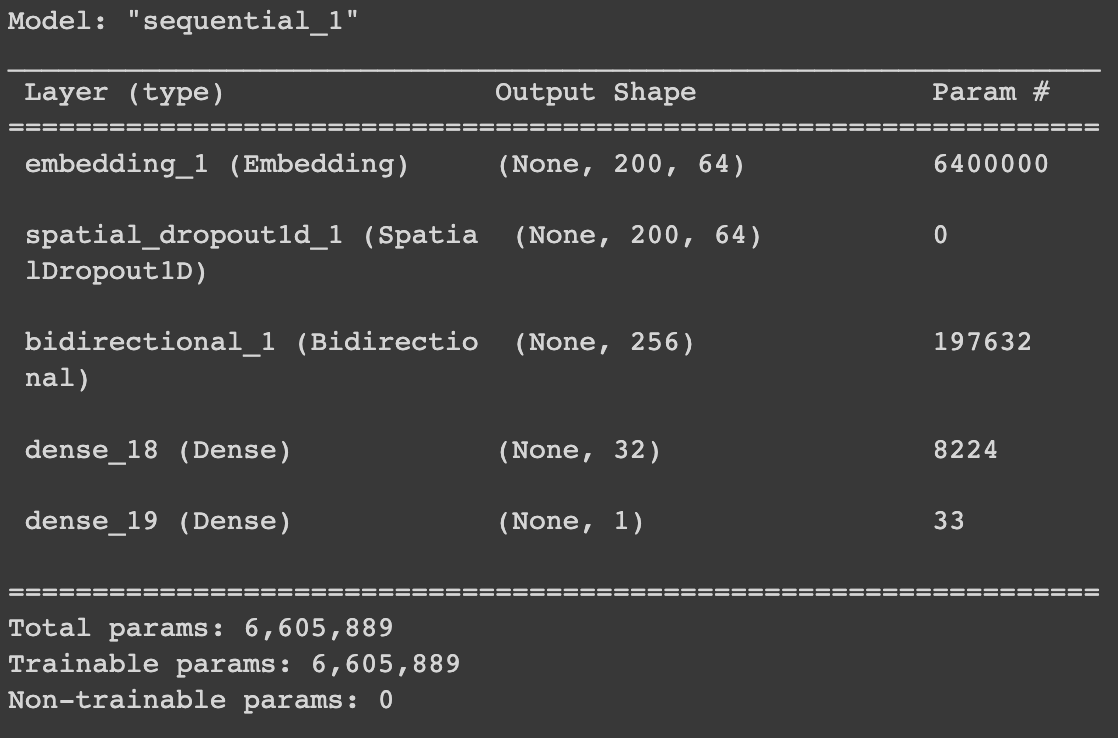
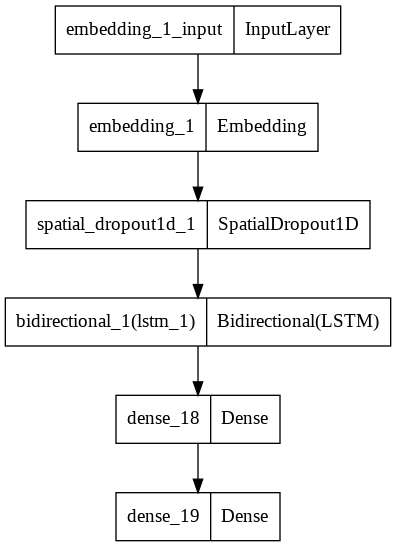
# IV. Methodologies

The models we considered for performing classification are **based on the Text Mining Document Term Matrix(DTM) file.** After generating the **DTM**, the columns are words and rows are descriptions of jobs, the value is the frequency of each word present in each job description. We use this file to build following models and calculate the accuracy, sensitivity and specificity of each model. Based on these results, we can evaluate the performance of our models for detecting fake jobs. We implement the supervised learning methods(Naive Bayes, K-Nearest-Neighbor, Classification Tree) and unsupervised learning methods (Association Rule) below to analyze the **DTM**. In addition, we use Bi-directional long short term memory in text mining.

* Bi-LSTM(Bi-directional long short term memory)



* Introduction: Bidirectional long-short term memory(bi-lstm) is the process of making any neural network who have the sequence information in both directions backwards (future to past) or forward(past to future). Therefore, it’s widely used in NLP problems, because human languages are context sensitive grammar.
* Used packages: Implemented in python, it uses TensorFlow, Keras, and especially Keras layer: Bidirectional & LSTM as core Bi-LSTM layer. We also used nltk and SpaCy, which are both open-source software library for advanced natural language processing, and chose SpaCy, the one that ran faster.
* Process:
  + Text cleaning: make text lowercase, remove special characters, links, remove punctuation and remove words containing numbers.
  + Remove stop words.
  + Lemmatize, which is also known as stemming, that is changing words to it’s root.
  + Tokenize text to sequences.
  + Build Bi-LSTM model, and model fit, as:

# V. RESULTS AND FINDING

The accuracy of linear regression and logistic regression based on the raw data (have not been processed to dtm format) is quite low, and cannot reach the accuracy, sensitivity of random guess. So we exclude them from our report and decide to build models after the text mining process.

1. Naive Bayes

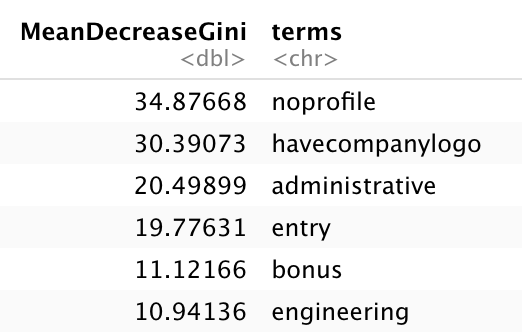
In text classification of Real/Fake job, our goal is to predict the conditional probability that fake job occurs in a document given that a word occurred in the document. To achieve this, we calculated P(term|fake) and P(term|real) and applied Bayes Rules. In addition, words in documents are assumed to be conditional independence.

In our model, accuracy is 0.8400, sensitivity is 0.8539 and specificity is 0.8393. Therefore, Naive Bayes model is pretty simple and efficient for handling such a large number of predictors (Over 700 columns).

1. Random Forest

Random Forest is one of the most important ensemble methods, which combine the results from multiple models with the goal of improving prediction accuracy. In text mining, we set the number of variables tried at each split equal to 27 (square root of total 715 columns) and generated an importance report.

Random Forest gives us the highest specificity which is 1, and we can combine this model with other models to vote for a prediction result.



From the importance plot of words, “noprofile” is most important to help us to classify a new record. “have company logo”, “administrative”,”entry” and ”bonus” are also more important than others.

1. Association Rule

we apply association rules to identify which words always occur together with “fraudulent”. Then we choose the rule with the highest lift and classify documents which include those words as “fraudulent”. We set minimum support=0.025 and minimum confidence=0.08.

In our model, the rule {nocompanylogo,noprofile,noquestions}=>{fraudulent} has the highest lift (lift=5,3322, confidence=0.2573, support=0.0264). We classify the documents which include “nocompanylogo,noprofile,noquestions” at the same time as “fraudulent”

1. K-Nearest-Neighbor

In the DTM, entries are the count of words in each document. In KNN, we classify a new document using the major vote of the classes of its K nearest neighbors.Greater distance indicates dissimilarity between documents, and smaller distance indicates co-occurrence of words in documents.

We choose kknn() to find the best K and predict the model, because using knn() to find best K takes too much time. Our best K is 2, and then we use it to prune the tree.The accuracy of KNN model is 0.9671, higher than Naive Bayes and Association Rule.

In addition, we also make a cross validation model using kknn. Accuracy is 0.9699, sensitivity is 0.7771 and specificity is 0.9797. Cross validation KNN performs quite well compared to others.

1. Classification Tree

In text mining, words are regarded as predictors to classify a new document as real or fake. We build a classification tree and prune the tree to find the best number of splits. The classification tree is shown in the appendix. Accuracy of classification tree is 0.9687, higher than Naive Bayes and Association Rule.

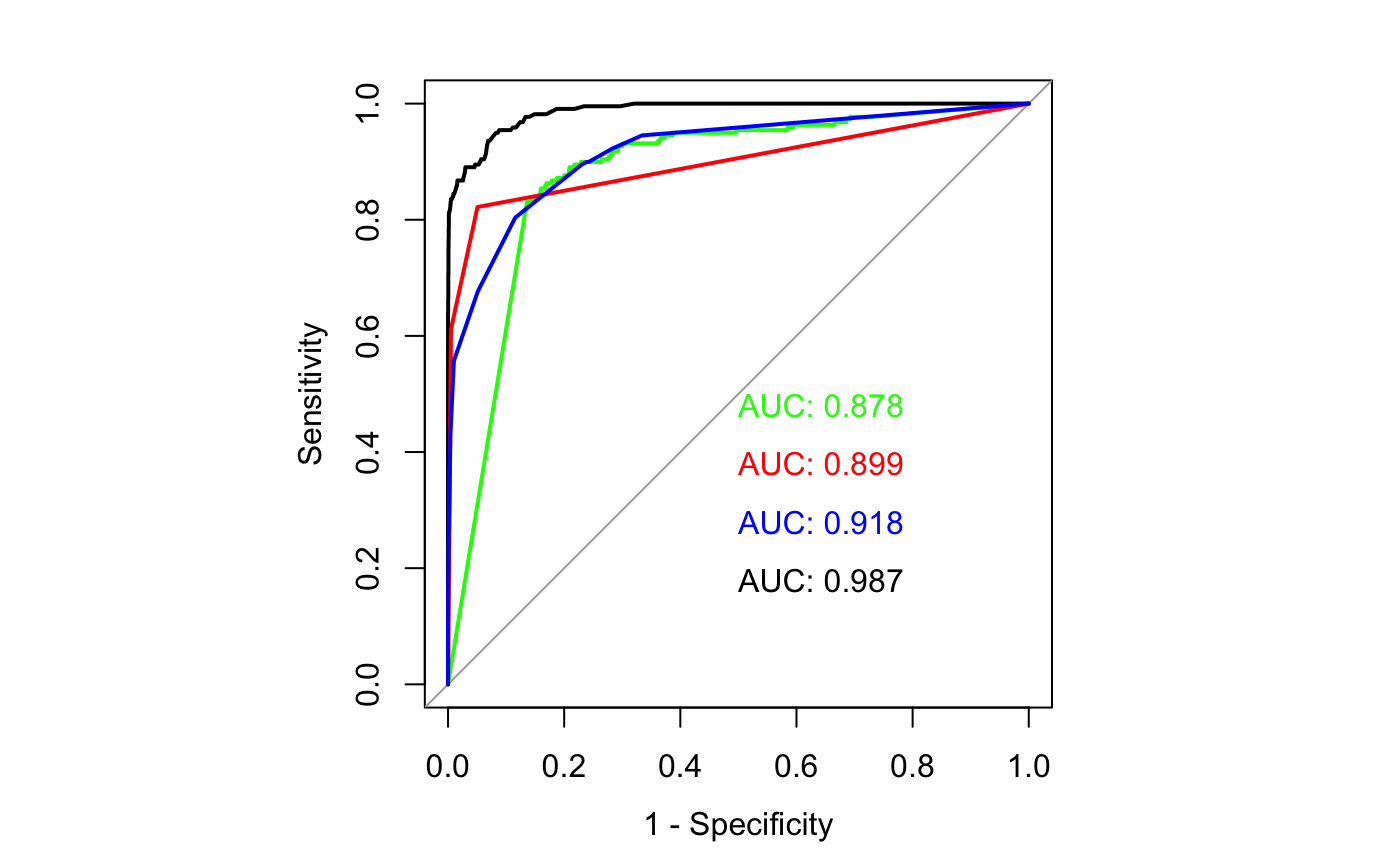
1. Accuracy, Sensitivity and Specificity

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **Sensitivity** | **Specificity** |
| **Naive Bayes** | 0.8400 | 0.8539 | 0.8393 |
| **Random Forest** | 0.9801 | 0.5936 | 1 |
| **Association Rule** | 0.9125 | 0.5799 | 0.9297 |
| **KNN** | 0.9671 | 0.7626 | 0.9777 |
| **Classification Tree** | 0.9687 | 0.5571 | 0.9899 |
| **Bi-LSTM** | 0.9803 | 0.6484 | 0.9974 |
| **Vote by NB, RF, Bi\_LSTM** | 0.9828 | 0.6941 | 0.9976 |
| **Vote by NB, RF, Bi\_LSTM, KNN, Tree** | 0.9841 | 0.7032 | 0.9986 |
| **Vote by NB, RF, Bi\_LSTM, KNN, Tree, Association Rule** | 0.9814 | 0.7717 | 0.9922 |

Based on the results in the table above, we can find that the 5-voter model has the highest accuracy, and the Naive Bayes model has the highest sensitivity, while Random Forest has the greatest specificity, which is 1. Let’s only focus on these performance measures, since we have no way of getting an objective cost matrix. Weighing these three metrics, we think the last 6-voter model has the best performance, because it has an outstanding accuracy value, it also keeps over 99% specificity, at the same time, its sensitivity is the second best among all of these models.

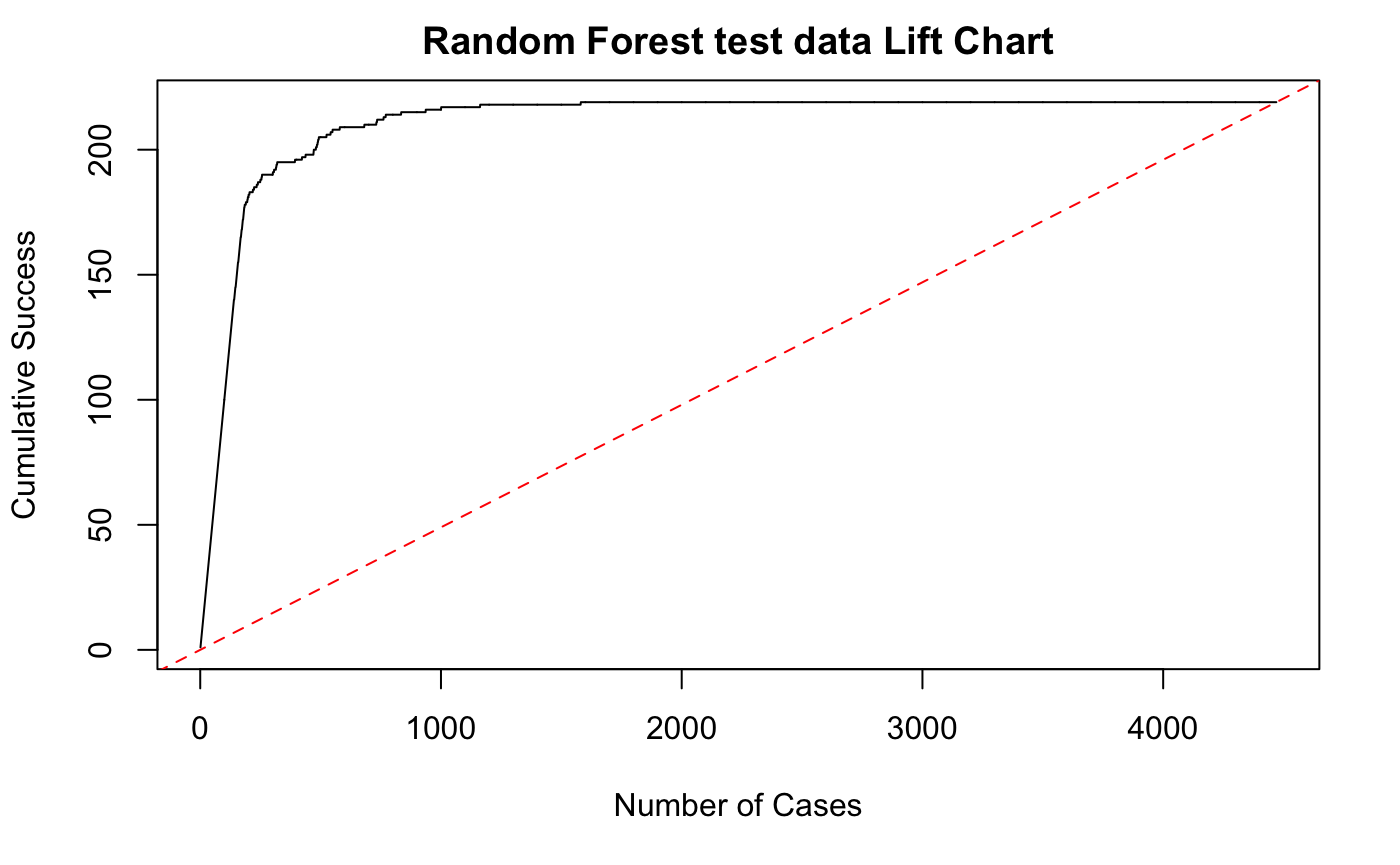
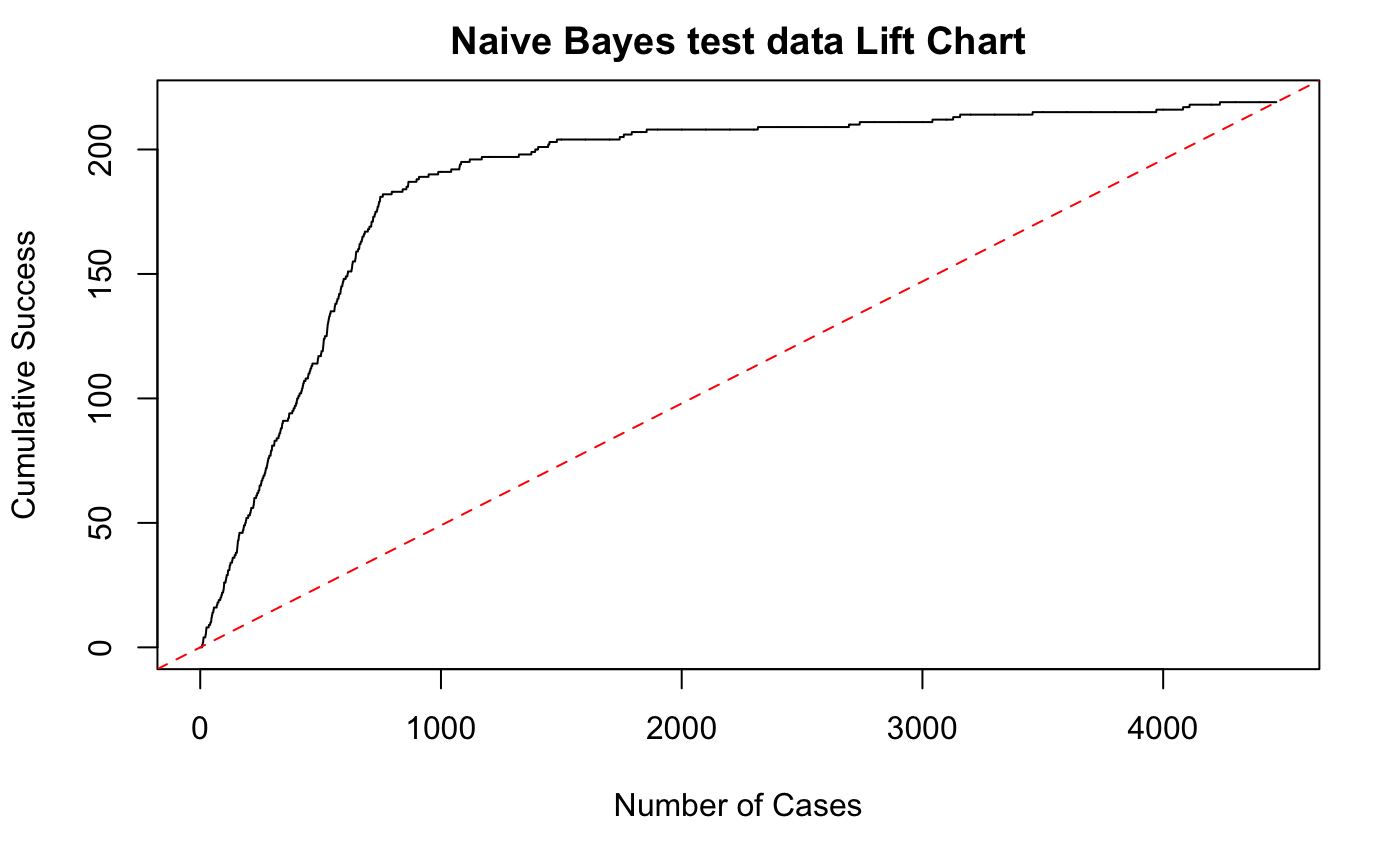
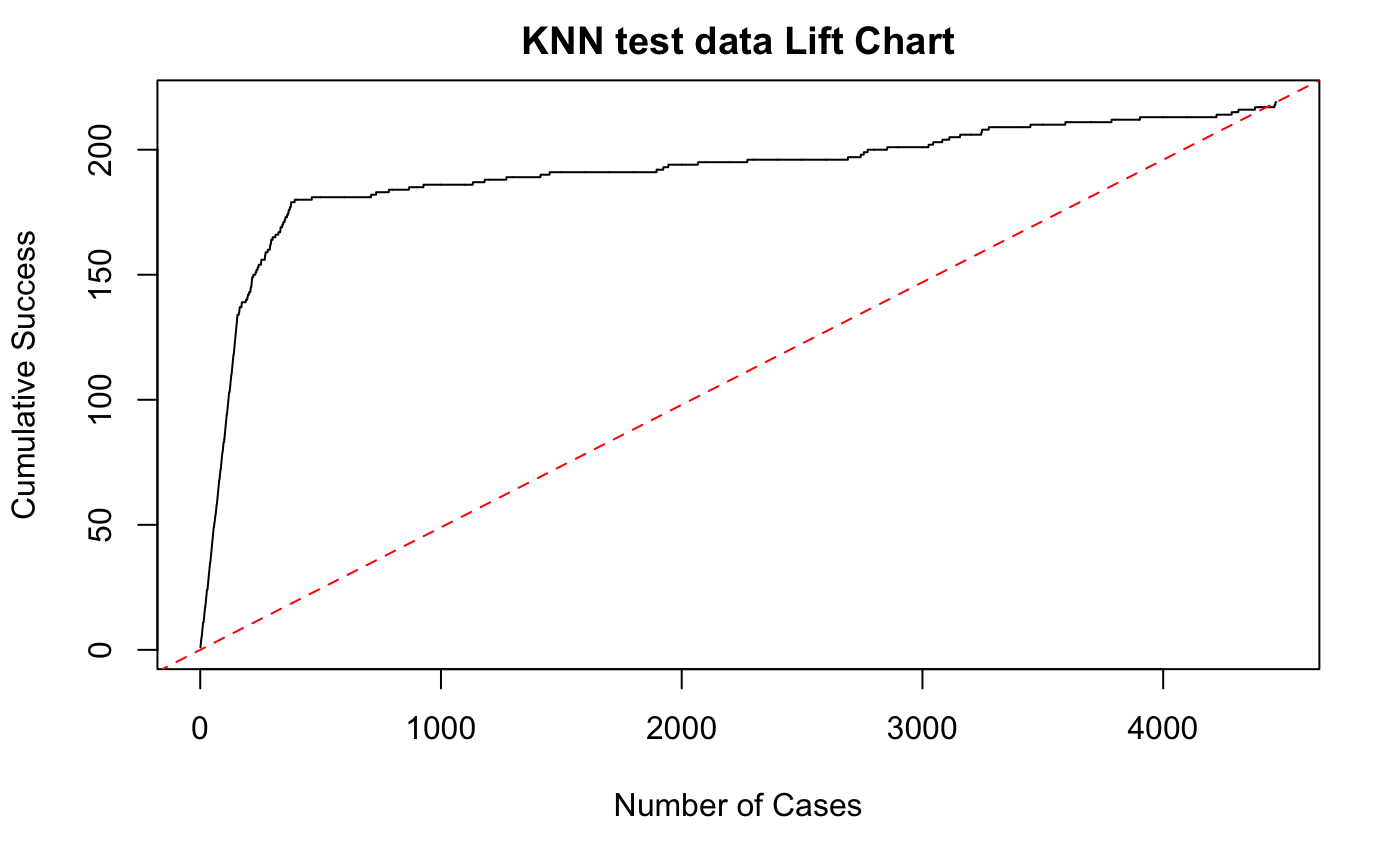
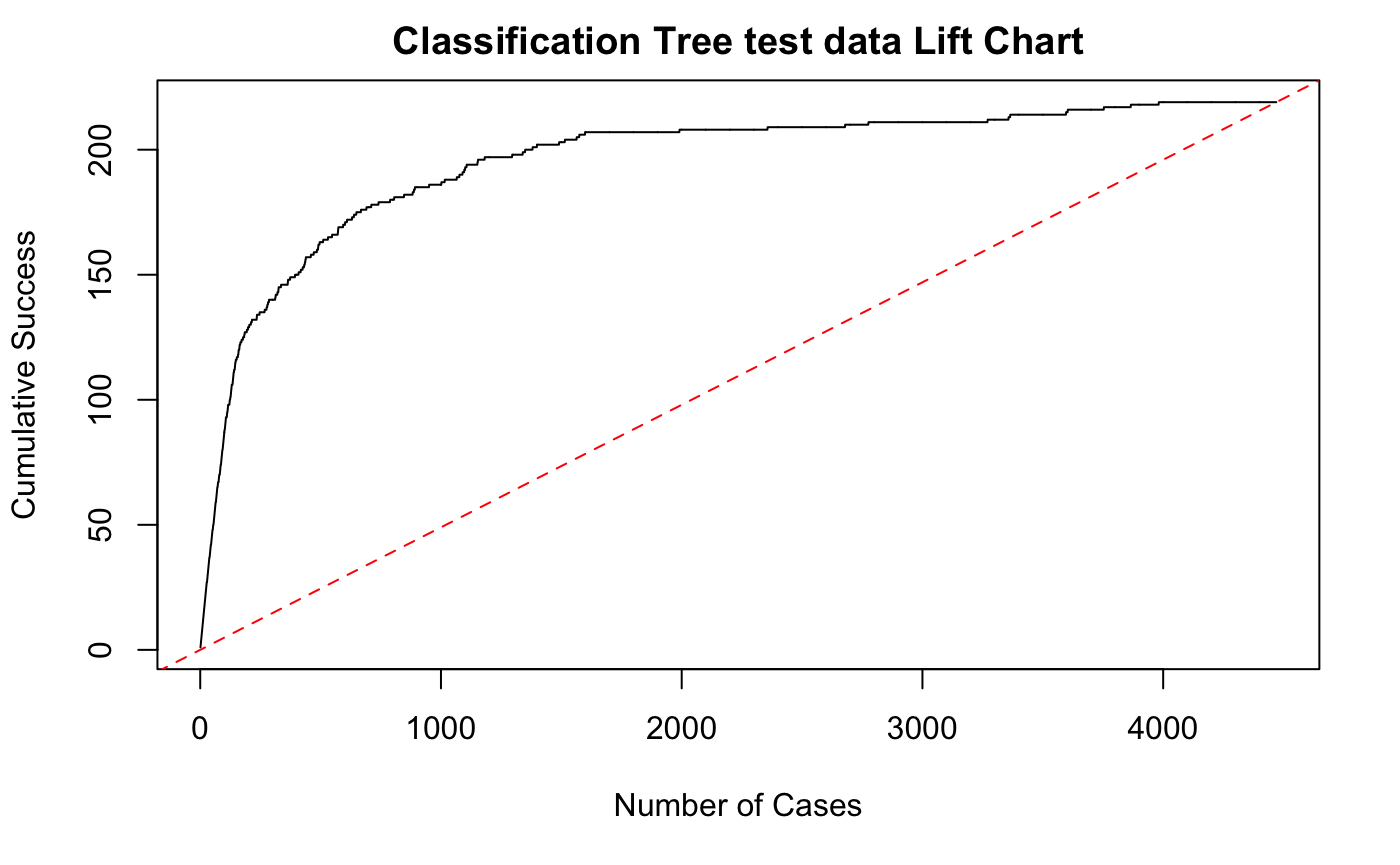
Moreover, if any researchers or analysis team has a fair cost matrix, they can combine our models with their cost matrix and evaluate which model has the lowest cost. I believe our model will present a soundness result.

1. Model comparison: ROC curves and Lift Charts



Black line: Random Forest Red line: KNN Green line: Naive Bayes Blue line: Classification Tree

As it can be seen in the ROC chart, the Random Forest model has the largest area, which means it has the best performance. While the Classification Tree has a poor performance among other models.

****

As can be seen, the Random Forest model has the best performance of the four models. In the lift chart, at the beginning, the cumulative successes increase rapidly,especially for the first 300 cases. The slope of that range is larger than that of the baseline, which represents random selection using naive rule.

# VI. APPLICATIONS

* Our model has a strong ability to detect fake jobs, with overall 98.4% prediction accuracy, and is able to find 70.3% of the fake jobs, but with confidence to claim that our reported jobs are mostly fake, by a proportion of 99.86%. Therefore, we claim our ensemble model to be a relatively strongly sound but weakly complete model.
* To change our prediction preference, e.g., input the cost matrix and minimize the misclassification cost, we could change the cutoff of our model, which is the minimum positive vote we receive from voters (models) that is enough for our model to assign positive.

# 

# VII. Future Work

* Packaged our model to a browser plugin.
* Implement web analytics to grasp information from job posting pages as posting records.
* Implement SQL to automatically store posting records into our database.
* Update our prediction model periodically, using our stored records.

# VII. REFERENCES

1. Dataset: <https://www.kaggle.com/code/tanvikurade/fake-job-postings-using-bidirectional-lstm/data>
2. https://en.wikipedia.org/wiki/TensorFlow
3. <https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=Bidirectional%20long%2Dshort%20term%20memory,forward(past%20to%20future)>.
4. <https://arxiv.org/abs/1508.01991>
5. https://towardsdatascience.com/using-the-missingno-python-library-to-identify-and-visualise-missing-data-prior-to-machine-learning-34c8c5b5f009
6. https://en.wikipedia.org/wiki/Scikit-learn
7. https://www.wikiwand.com/en/Logistic\_regression
8. https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/