# <u>Fake Jobs Detector Algorithms</u>

*M00724895* *Mei Tan Le Ping*

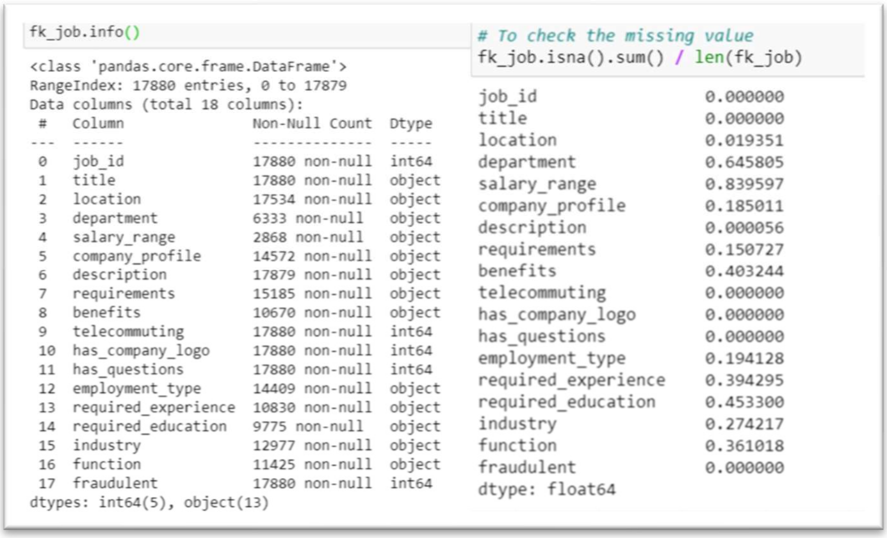
## 1. Introduction of the Dataset and Machine Learning problem

Fake jobs defined as employment scam which attempt to defraud people who are seeking vacancies by tricking victim's good employment with unrealistic hope but the vacancy is none-exist. Previously, jobs advertised was via newspapers, TV or Radio, but in the era of technology more jobs are posted online. According to B. Alghamdi et la (2019) reports, job scam has become one of the criticl problems nowadays which had been addressed in the domain of Online Recruitment Frauds(ORF). This has drawn attention of worldwide but there are less studies found on it. Scammers understand that it is challenging of getting a job now especially in the Covid-19 pandemic which had lead to open the door for scammers by offering attractive salary, favourable task functions and hours with lower requirements of qualifications, skills or experiences, etc. These fraudulent jobs posting online normally will be found on the same website of the genuine company with same name and logo posted vacant positions, while later scammer would request applicants to furnish personal details, passport details, credit card or banking information and also to remit payment for miscellaneous type of upfront fees such as background check, training kits or even processing fees in the exchange of securing the job.

According to a survey carried out by Safer jobs which is an organisation help to combat employment fraud in United Kingdom stated that there are more than two thirds of people are now seeking employment online. However, 98% of the respondents stated that they tried to believe and applied online job even though they have suspected scam and not legitimate. There are few types of employment scams upon application such as mystery shopper, multi-level marketing, guaranteed income or job, payment administration, visa or permits processing fees and more. All above happened where scammers would manipulate and required applicants to provide more details and money. As Copper et la (2018) reports, fraud is defined as the cheating for financial gain and value of United Kingdom with roughly £110 billion per annum. International trade violence and fraudulent are rising alarming threating the economies with the fraudulent jobs. In addition, this crime activities might be the newest method of money laundering.

Due to the rise of job scams, machine learning on prediction of employment scam started getting popular to be studies these days where many people are unemployed or work at home during COVID-19 pandemic. Better features of machine learning of classification algorithms will be study to predict if a job posted is fraudulent or genuine. To further research on the important features of fake jobs such as company profile, description, requirements and other details based on the texts. Exploratory analysis of visualisation with be carried out for a better understanding the data based on the length pattern and insights of genuine and fake jobs. Data cleaning and processing on missing value and transformation were executed later.

Jupyter notebook in Python was being implement for the text analysis and text mining project and the datasets used was drawn from Kaggle [3] with the training dataset has 17880 rows and 18 columns of features which comprises of texts mainly. Features selection and pre-processing of the raw datasets was carried out thoroughly in order to train models. Figure 1.1 is the details of the 18 attributes, which showing the data types (left side) and missing value in percentage (right side). Percentage of missing value used for decision to delist of columns with higher percentages prior the process of data cleaning of filling up missing value.

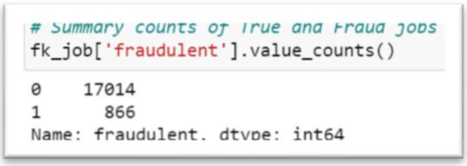


```
fk_job.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   job_id               17880 non-null  int64 
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64 
 10  has_company_logo     17880 non-null  int64 
 11  has_questions        17880 non-null  int64 
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64 
dtypes: int64(5), object(13)
```

```
# To check the missing value
fk_job.isna().sum() / len(fk_job)

job_id                 0.000000
title                  0.000000
location               0.019351
department             0.645805
salary_range           0.839597
company_profile        0.185011
description            0.000056
requirements           0.150727
benefits               0.403244
telecommuting          0.000000
has_company_logo       0.000000
has_questions          0.000000
employment_type        0.194128
required_experience    0.394295
required_education     0.453300
industry               0.274217
function               0.361018
fraudulent             0.000000
dtype: float64
```

Figure1.1 Structure of dataset

There are total of 17 independent variables, whereas 'fraudulent' attributes as the dependant variables for this text analysis machine learning project. Fraudulent labelled with Boolean value, 0 indicated genuine job; 1 indicated fake job. There are total of 17014 counts of genuine whilst 866 counts of fake jobs as shown in figure 1.2.



```
# Summary counts of True and Fraud jobs
fk_job['fraudulent'].value_counts()

0    17014
1      866
Name: fraudulent, dtype: int64
```

Figure 1.2 Count of label classes

## 2. Literature review of relevant machine learning techniques

Research on fake job detection, email spam detection, fake news detection, review spam detection are some of the popular studies in the domain of Online Fraud Detection. According to E. G. Dada et la (2019) studies on the machine learning problems, they noted that unneeded mails usually got into the user mailbox are spam mails which has lead to unavoidable storage and consumption issues. Thus, Yahoo mail, Google and Outlook service providers implement Neural Network to spam filters to eliminate this challenges. While N. Hussain et la (2019) done another research on the Spam's review detection. Spa review content would manipulate the sales of products and bring profits. Their studies was developing techniques and algorithms by extracting features from the review spam using Natural Language Processing (NLP). For fake news detection of studies of K. Shu et la(2017) trying to find the 3 reason how it was written, how it was spread and who is was relating to by extracting the features on the news and social content by developing a machine learning algorithms.

All above issues are approached by developing machine learning algorithms of classification by recognising the fraudulent label. Classification as a supervised learning algorithms with classifier map the input of independent features or data to target the classes label in training data which later will be test on test data for accuracy result of the models. Classification is trained to forecast the unknown test data, the common algorithms for Single based are Nave Bayes Classifier, K-Nearest Neighbours , Decision Trees and Multi-layer Perceptron.

I.Irish (2014) had mentioned that Naïve Bayes is a supervised classification that the accuracy of model is not highly dependent on the features instead of the amount of the data, while D.E Walters (1988) said that it classify the data based on conditional probability. I is used to predict fraudulent label as it is scalable and apply to predict label of texts which is suitable for text classification challenges via calculation of the probability of every single text and then return it as it has the highest probability with promising result.

In a report of P. Cunninghm et la (2007) has stated that K-nearest Neighbor classification known as a lazy algorithms by identify the objects on the closet proximity of training data to forecast the classes and label by determining the best k value.

Decision Tree classification is defined as classifier with learning data with the use of tree structure as studies of H. Sharma et la(2016). The leaf node is denoted to every label class whilst the non leaf node will be used to indentify the test data as decision node. The result will be predict by the branches of decision node.

Text Classification will be undergoing text mining, Natural language processing (NLP) and analytics. Text mining is a process of processing text data itself and understanding the pattern and sizeable of the textual data. While, NLP is the stage of underlying metadata. Later stage follows by reading the counts of frequent used words, length of the text, existence of specific words is the process of text analytics. Text mining is the initial stage of the whole process which pre-processed the datasets and ready for the next stage of text analytics and machine learning algorithms for information classification.

NLTK is one of the Python packages that is powerful to provides natural language algorithms that assist the machine to understand, analysis, pre-process the text data. In addition, it is free and open source and well documented. NLTK comprises of the popular algorithms such as stemming, topic segmentation, part of speech tagging, tokenizing and more. Tokenization is the initial stage of text analytics which will break down a sentence into smaller chunks of words or sentence name token. There are three types of tokenization which are sentence tokenization, word tokenization and frequency distribution. Next will be the stopword module which responsible for removing noises in texts such as is, am, a, an, the, etc. Lexicon Normalization is a process of reducing the noises in texts derivationally related form of words to a common root word. While, stemming is another module of NLTK which is the normalization of linguistic by reducing the words into the basic word. A more advanced module than stemming called lemmatization which is a process of reducing words to their base word by transforming root word with the use of vocabulary and morphological analysis with the help of dictionary look-up. Part-of-Speech (POS) tagging is another package which identify the grammatical group of text via analysis of the relationship within paragraphs by assigning respective tag to the word.

Text classification is a supervised approach and one of the crucial stages in text mining. It is widely applied in all industries these days to detect spam, products or tasks categorization, website content categorization, etc. The classification methodologies initiated with data loading and exploration, follow by pre-processing such as features engineering to improve accuracy and then split train and test set function for models building. Performance of models built will be evaluated.

## 3. Machine learning pipeline adopted

The objective of this project is to predict if a job posting is fraudulent or genuine. Firstly, dataset was downloaded [3] in csv file and it was loaded into server of Jupyter notebook for analysis via python. This original dataset will be used to train and test the models built for the best result of machine learning algorithms. Due to the imbalance and missing value of the original datasets, cleaning and pre-processing techniques was carried out to balance the dataset by removal of missing values, removal of irrelevant attribute, elimination of extra space and stopwords. Basically, missing values were handled, and the trailing spaces accounted for and text was converted into vectors to be ready as finalise datasets for feature selection and models building. Pandas and Numpy are the most important libraries needed for this process.

Secondly, feature selections of impactful and important features to detect the fake jobs posted. Features of company profile, description and requirements are deemed to be heavily impactful on the authenticity of a job posting. This leaded to the need of adding new column of combining these three features of text for further analysis and machine learning pipeline. Sklearn Machine Learning Libraries are needed in these stages.

Evidently as figure 3.1, text length pattern shows similar sign for both genuine and fraudulent jobs on the behaviours attributes of company profile, description and requirements. These three features will be combined into the same column for ease of text analysis.

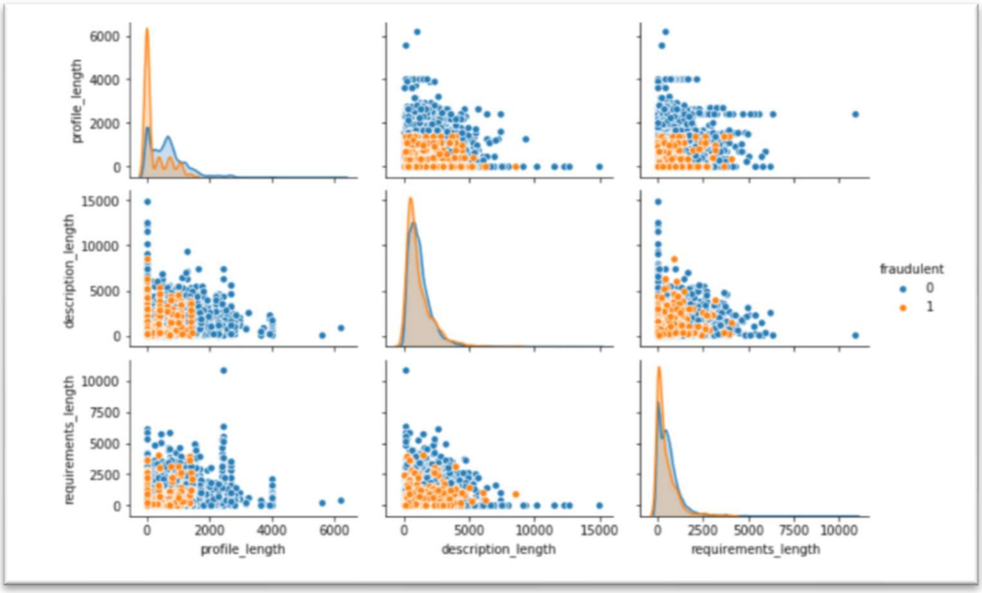
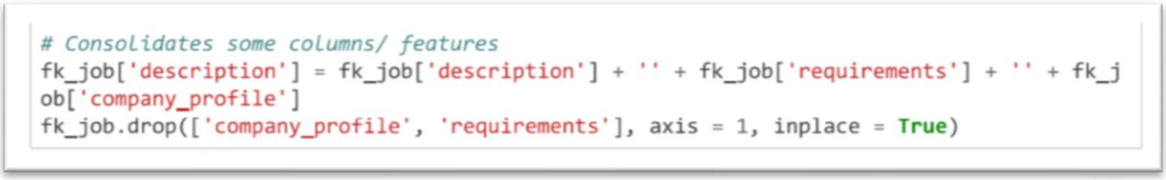


Figure 3.1 Relationship of text length

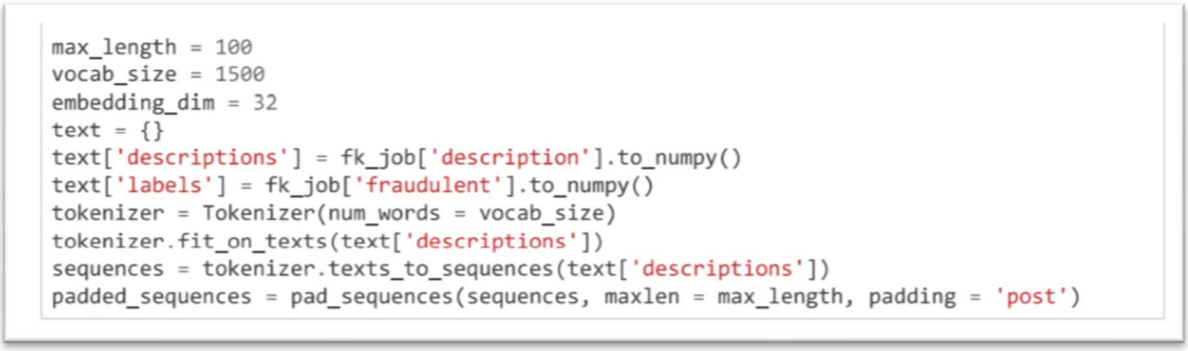
Text analysis of this project was analysed by created wordcloud upon pre-processing of data where later followed by tokenisation with the help of stopwords to remove punctuation and signs. Charts and histograms are later to visualise the analysis and comparison. Spacy documents was created to tokenise the dataset’s text. The selected features of three column of company profile, description and requirement have been combined into a column to be analyses as Figure 3.2 below.



```
# Consolidates some columns/ features
fk_job['description'] = fk_job['description'] + '' + fk_job['requirements'] + '' + fk_j
ob['company_profile']
fk_job.drop(['company_profile', 'requirements'], axis = 1, inplace = True)
```

Figure 3.2 Combined of features into Description column.

Tokenize process as shown in Figure 3.2 below. This is by splitting each word into a separate token which is also know as element in a list. Removal of punctuation and stopwords in this processing will be executed. Analysis of common and frequency of words can be studies onwards. While the next Figure 3.4 is showing the most frequent words used in fake job.



```
max_length = 100
vocab_size = 1500
embedding_dim = 32
text = {}
text['descriptions'] = fk_job['description'].to_numpy()
text['labels'] = fk_job['fraudulent'].to_numpy()
tokenizer = Tokenizer(num_words = vocab_size)
tokenizer.fit_on_texts(text['descriptions'])
sequences = tokenizer.texts_to_sequences(text['descriptions'])
padded_sequences = pad_sequences(sequences, maxlen = max_length, padding = 'post')
```

Figure 3.3 Tokenize the combined description text

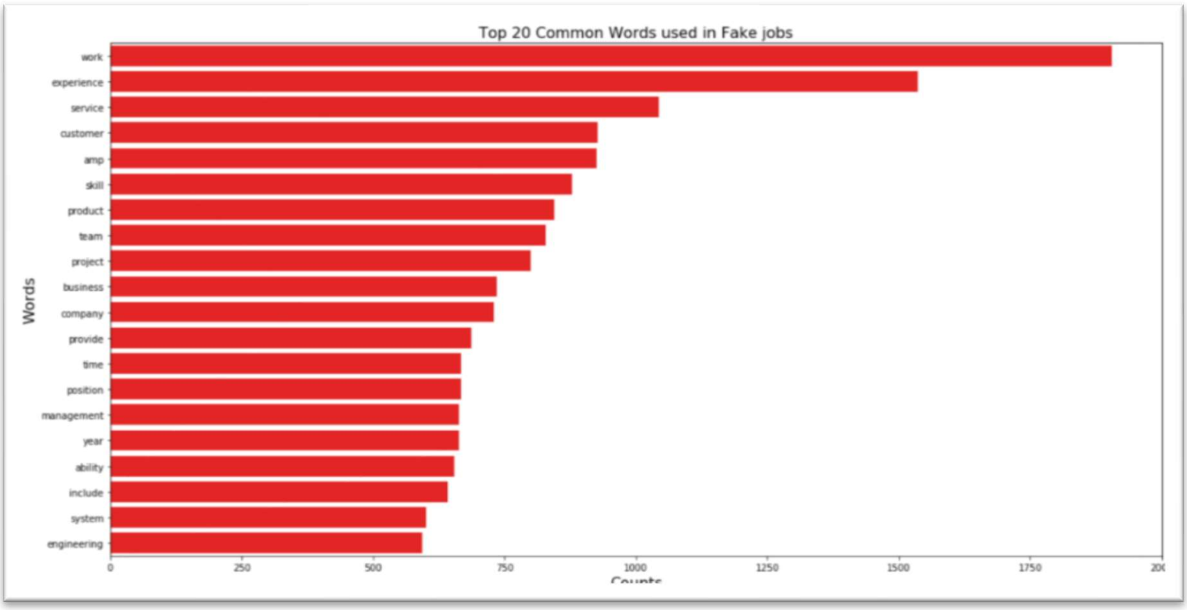


Figure 3.4 Top 20 words common words in fake jobs

Upon exploring and understanding of data and features selection which followed by data transformation via text mining. The dataset prepared is in vector form will be used to train and test on 5 classifier for forecasting the best results. The classifier used were Gaussin Naïve bayes classifier, Logistic Regression classifier, Random Forest classifier, K-Nearest neighbours classifier, Multi-layer Percepton (MLP) classifier for classifying job posted as fraudulent. The target label of fraudulent column serve as the dependent features for classification objective.

Firstly , the processed dataset is split into 75% train set for models training, whilst the remain 25% used for prediction accuracy as shown below Figure 3.3. The performance of every single classifier are evaluated base on metrics area under the Accuracy, ROC Curve (ROC_AUC) and F-measure. The best performance text classifier will be chosen as the best model of machine learning.

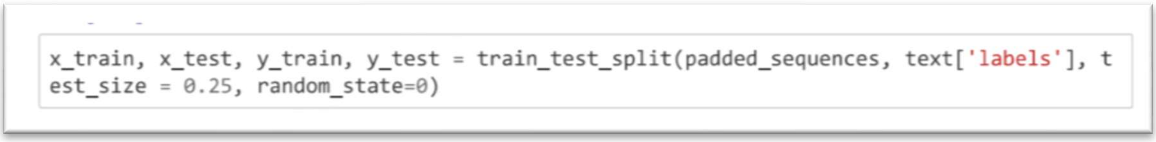

```
x_train, x_test, y_train, y_test = train_test_split(padded_sequences, text['labels'], t
est_size = 0.25, random_state=0)
```

Figure 3.3

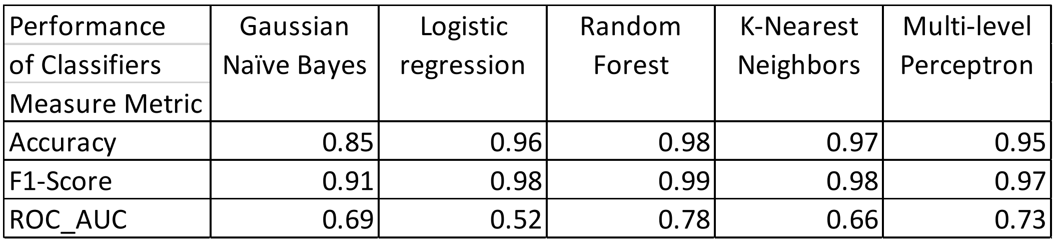
For training modelling, all the text classifiers except Naïve Bayes classifier has been tuned with appropriate parameters to optimise the performance results due to the fact that default parameters might not be suitable for every problem-solving machine learning algorithms. Gaussian Naïve Bayes is in default state and serve as the baseline modelling. Tuning of the parameters will increase the reliability of the machine learning algorithm by detecting the job post as fake.

Logistic regression was tuned with c-values 10 and penalty options 12 to obtained the best score of performance metrics to be compare among all other models of classifiers. While Random Forest classifier was adjusted for the parameters of n_estimator to 200 to optimise the result. K-nearest Neighbors with the best performance with k value of 5. Last but not least the Multi-lever percepton with solver 'lbfgs'. All the training data was fitted into the adjusted model followed by test data fitted to obtained the results for prediction purpose. Machine Learning models will be evaluate based on the actual and predict values to identify the best model which is problem solving.

Performance of all the classifiers models were justified for evaluation. As per reports of H.M et la (2015), he mentioned that Accuracy is a metric to find out the ratio of prediction of true value over the total instances. However, this metric would not be the best metric to evaluate performance of models due to the fact that it does not predict the wrong value. Moreover, false positive and false negative is necessary to be considering as well to reduce the inaccurate classification. Recall and precision would need to be consider as well to reduce the misclassification of modelling. Recall is to get the ratio of true positive outcome over the total instances; precision is to identify the ratio of true positive over the total of positive output. F-measure is a metric that identify the both precision and recall and calculate the mean among two measurements. The area under the ROC curve is a metric to measure the whole dimensional area below ROC which is the classification threshold invariant. The higher the score of Accuracy, F-measure and ROC-AUC indicating the better the model performing.

## 4. Evaluation against baseline technique

Scikit-learn of Gaussian Naïve Bayes classifier was implemented in order to get a baseline accuracy rate for this project. Naïve Bayes is known to be the simplest approaches classification which probabilistic is implemented with handling of all the features selected are independent to the class labelled. The parameter of Naïve Bayes classifiers was default and used to compare with another 4 classifiers models performance as shown in Figure 4.1



| Performance of Classifiers Measure Metric | Gaussian Naïve Bayes | Logistic regression | Random Forest | K-Nearest Neighbors | Multi-level Perceptron |
|---|---|---|---|---|---|
| Accuracy | 0.85 | 0.96 | 0.98 | 0.97 | 0.95 |
| F1-Score | 0.91 | 0.98 | 0.99 | 0.98 | 0.97 |
| ROC_AUC | 0.69 | 0.52 | 0.78 | 0.66 | 0.73 |

Figure 4.1 Classifiers Performance Comparison

Above table is the experimental results of this project and it does not shown much difference in the measurement metric of Accuracy and F1-score (F-Measure) among all the classifiers. Most of the models showing similar measurement and it is hard to decide which is the promising machine learning algorithms to predict fake jobs posting. However, in the third rows of metric measurement of Area under ROC(Figure 4.1) clearly shown that Random forest classifier is providing the best performance among all the 5 classifiers. Moreover, the baseline model of Gausian NB classifier scored lowest among all the models built.

Figure 4.2 below is a graph comparing the ROC_AUC results of all the classifiers tested for genuine and fake jobs predictions. Random Forest scored highest with performance of 78%. Specialy to highlight that, Random Forest classifiers with n_estimator of 200 also scored the highest for both the measurement metrics in term of Accuracy and F-measurement among all other classifiers. Random Forest classifier shown an excellent and significant performance models. The second highest performed classifier model multi-level perceptron with 73% of ROC_AUC score and 95% of accuracy and F1-score of 97% which is competence to random forest classifier.

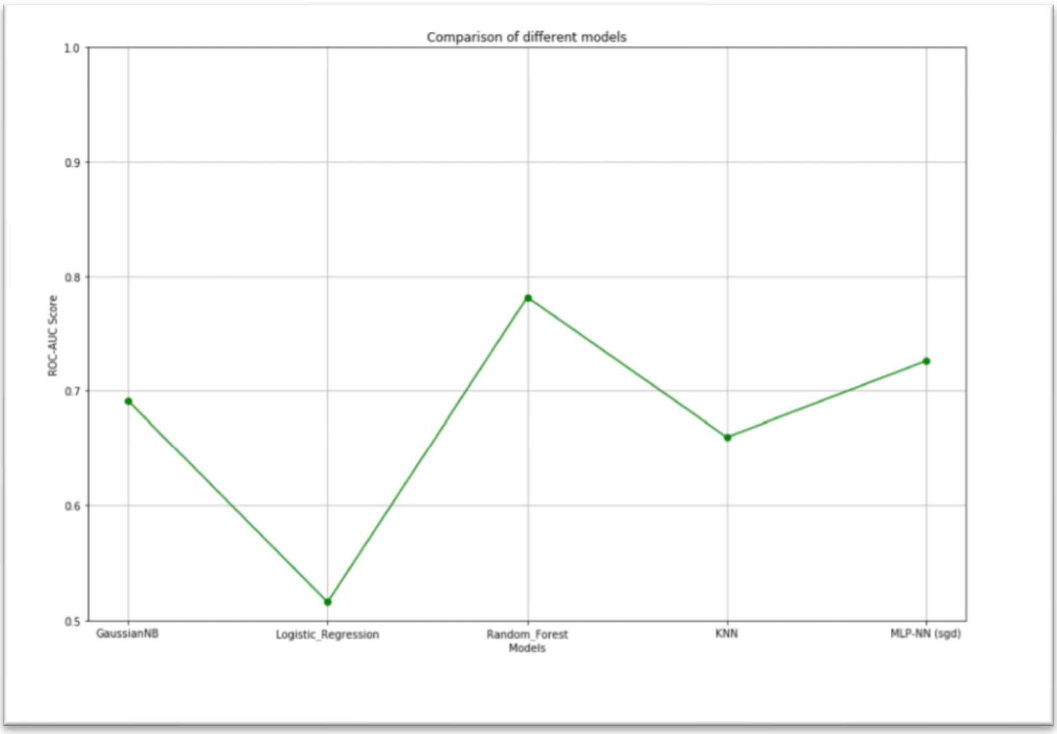


Figure 4.2 Comparison of ROC_AUC Score 5 Classifiers

## 5. Conclusion

Genuine and fake jobs posting detection will reduce job scam and bring losses to victims. Job seekers will only lead to the genuine jobs posting and legitimate employer. This project had selected the relevant and best features of text to predict the fake jobs. The features selected are the company profile, descriptions and requirements text column to undergone text analysis and fitted into 5 classifiers models which is adjusted with appropriate parameters except Gaussian Naïve Bayes with default parameter which is baseline techniques for models performance comparison. Final outcome of this experiment shown that Random Forest classifier furnish the most promising performance in all the 3 measurement metrics applied. Random Forest classifiers with n_estimator of 200 achieved 98% in predicting fake jobs posting.

## References

1. B. Alghamdi and F. Alharby (2019). *An Intelligent Model for Online Recruitment Fraud Detection.* pp. 155–176

2. Copper, Christopher (2018). *"Fraud epidemic costs the UK £110bn annually: report". www.internationalinvestment.net.* Retrieved 2019-08-20.

3. Datasets, *Kaggle, https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction,* February 2020.

4. E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, (2019). *Machine learning for email spam filtering: review, approaches and open research problems.* vol. 5, no. 6

5. N. Hussain, H. T. Mirza, G. Rasool, I. Hussain, and M. Kaleem, (2019). *Spam review detection techniques: A systematic literature review*. vol. 9, no. 5, pp. 1–26.

6. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. (2017) *Fake News Detection on Social Media*. Newsl. vol. 19, no. 1, pp. 22–36

7. I. Rish. (2014) *An Empirical Study of the Naïve Bayes Classifier An empirical study of the naive Bayes classifier.* no. January 2001, pp. 41–46

8. D. E. Walters. (1988) *Bayes's Theorem and the Analysis of Binomial Random Variables, Biometrical J.,* vol. 30, no. 7, pp. 817–825

9. P. Cunningham and S. J. Delan. (2007) *K -Nearest Neighbour Classifiers, Mult. Classif. Syst.*

10. H. Sharma and S. Kumar. (2016) *A Survey on Decision Tree Algorithms of Classification in Data Mining.* vol. 5

11. H. M and S. M.N. (2015) *A Review on Evaluation Metrics for Data Classification Evaluations.* vol. 5, no. 2