Department of Electrical and Computer Engineering
North South University

Senior Design Project

Fake Job Posting Detection Using Machine Learning

Tawfiqul Islam Talukder **ID # 1521092042**

S.M Shouvik Islam **ID # 1712767642**

Faculty Advisor

Dr. Mohammad Monirujjaman Khan

Associate Professor

ECE Department

Fall, 2022

DECLARATION

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' name & Signature

1. **Tawfiqul Islam Talukder**

2. **S.M Shouvik Islam**

APPROVAL

The capstone project entitled "**Fake Job Posting Detection Using Machine Learning**" by **Tawfiqul Islam Talukder (ID#1521092042)** and **S.M Shouvik Islam (ID#1712767642)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on May and has been accepted as satisfactory.

Supervisor's Signature

Dr. Mohammad Monirujjaman Khan
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair's Signature

Dr. Rezaul Bari
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

ACKNOWLEDGMENT

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The capstone project program is very helpful to bridge the gap between the theoretical knowledge and real life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjaman Khan for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 499 in which we could really work on this project and materialize it the way we have dreamt of. We thank our friends and family for their moral support to carve out this project and always offer their support.

ABSTRACT

This report presents the design and the implementation of a system that can detect fake jobs using a machine learning method that employs a variety of categorization algorithms. The COVID-19 epidemic situation has transformed the regular livelihoods of mankind in the world. This epidemic has put excessive pressure on the job market. As a consequence of the epidemic, most organizations have halted their recruiting processes, which has raised the rate of unemployment. Online recruiting has suddenly increased the quantity of applicants while also bridging the distance between recruiters and candidates. It indicates that scammers have emerged in the online recruiting market. They provide extremely high pay ranges or any other type of benefit on several online platforms. It's called "Fake Job Postings." Job seekers are applying for those fake jobs. As a result, scammers steal their personal information. Scammers use their personal information for a variety of cybercrimes or sell it on the dark web. This paper's objective is to identify and verify these job advertisements, whether they're fake or not. To identify these fake job advertisements, Machine Learning Algorithms (MLA) was implemented throughout this study, such as the Random Forest algorithm, and Logistic Regression algorithm. This study trained and tested the dataset and got an accuracy of 98.86 percent in Logistic Regression and 98.54 percent in Random Forest. In the Logistic Regression algorithm, our recommended technique has an accuracy of 98.86 percent, which is a huge improvement over the current methods. The accuracy percentile of both the algorithms used throughout this analysis is substantially in excess of prior studies, showing that the algorithms utilized throughout this analysis are well balanced.

Table of Content

List of Figures

List of Tables

CHAPTER 1:
Introduction

1.1 Introduction

The increasing usage of the internet has greatly simplified the hiring process. Besides, the current pandemic has had a significant effect on the current shift in job recruitment trends. Online recruitment process has increased the number of candidates available and simplified processes, this has helped to bridge the gap between recruiters and potential prospects. Candidates may now apply to a large number of jobs depending on their specialty on the internet with a single click. On different online platforms, employers post job openings with the skills they seek. These websites allow job searchers and applicants to post their resumes and skill details. On different online platforms, employers publish job vacancies with the skills they seek. These websites allow job searchers and applicants to post their resumes and skill details. Companies check the backgrounds of prospective individuals and approach them, while job seekers can now apply for the job profiles that interest them. Organizations approach the selected individuals for additional procedure and hire the best prospects after the initial screening. Online hiring is advantageous to both candidates and companies. In recent years, scammers have sprung up in the internet recruitment market, resulting in a new type of scam known as "Online Recruitment Fraud." Fraudsters use online recruitment fraud to entice candidates with attractive employment offers while stealing their money and personal information. As a result, organizations' reputations are harmed, and job seekers' perceptions of the company are negatively influenced. Detecting fraudulent work proposals within a genuine variety of employment offers is a difficult challenge to solve technically. Main issue is the problem of class disparity, even as the number of fraudulent positions is low in comparison to legal jobs. This makes it difficult for automated systems to learn the characteristics of fraud jobs. Besides, some of the jobs being advertised are only phantom jobs set up to steal potential data [1]. When candidates apply for these positions, their prospective data is taken or, in some cases, their PCs are hacked to steal vital information. Even after years have gone, cybercriminals combine the victim's data and deliver it on the dark web for someone else to use. It is a challenging technological effort to detect fraudulent job offers within a legal collection of job offers. Because the number of fake employment is relatively modest in proportion to real jobs, the main issue is one of class inequality [2]. This discovery of fraudulent job postings has prompted a lot of interest in creating an intelligent

technique for identifying bogus jobs and warning individuals so they don't apply for them. Because the machine learning technique uses a variety of classification algorithms to detect bogus jobs, we will use machine learning to detect false job postings. Fake jobs are identified from a bigger number of job posts using a classification algorithm that alerts the user [3].

One of the most important challenges in recent years is the fake job posting scam. A fake job posting is a form of scam that is well-designed and targeted at job searchers for a number of unethical reasons. Scammers deceive individuals in believing that they have or will have a job. Criminals utilize their status as "employers" to persuade victims to provide them personally identifiable information, act as unwitting money exchangers, or pay them money. These scams have become easier and more profitable as technology has advanced. By impersonating company websites and advertising false job postings on prominent online job forums, cyber thieves are increasingly posing as legitimate employers. They stage fake interviews with naive application victims, then demand personal information and/or money from them. The PII can be used for a variety of criminal objectives, such as taking over the victims' accounts, opening new financial accounts, or exploiting the victims' identities in another deception scheme (for example, getting fake driver's licenses or passports).

1.2 Project Description

This section provides an overview of materials and tools, a dataset description and a description of the machine learning model used to determine whether job advertisements are fake or real. The dataset was gathered from the open-source platform Kaggle. For the detection of fake job postings, we applied some machine learning models, which are Multinomial Naive Bayes, Passive Aggressive Algorithm.

1.2.1 Materials and Tools

Almost all programming languages can be used to write machine learning-based applications, but the ideal programming language for data analysis is Python. Because of Python’s board library access, machine learning-based difficulties are particularly successful with python programming. Anaconda Navigator, Jupyter Notebook, and Google Colab were utilized to manage big datasets and perform model training online while using a personal GPU for dataset preparation.

1.2.3 Dataset Description

To detect fake jobs, we trained and tested the dataset, which was collected from Kaggle. The dataset consists of 18 columns, which has one unique id column, four integer columns and thirteen string columns and 17880 rows of textual and numerical data for training and testing machine learning and natural language processing (NLP) algorithms. These columns correspond to the titles and information seen in various job postings on sites like Indeed, Bd-jobs and others. These datasets provide a detailed picture of how job advertisements are posted online. Table 1 shows the sample of the columns, that is what kind of data are in the dataset.

Column Name	**Columns Description**
Job Id	Unique Job ID for all job post.
Tittle	The tittle of the job ad entry.
Location	Geographical Location of the job ad.
Department	Corporate department (e.g., sales).
Salary Range	Indicative salary range (e.g., $50,000-$60,000)
Company Profiles	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	Job posted company has his own logo. It represents with 1and 0.
Has Question	It represents have some question.
Employment Type	Job types represent what the job is its full time or part time etc.
Required Experience	It represents what kind of experience applier needed
Required Education	It represents what is the education level applier have
Industry	it represents which industry jobs
Function	It represents which function jobs
Fraudulent	it represents Job post is fake or real

Table 1.1: The sample of the columns

This dataset is utilized in the suggested approaches to evaluate the approach's overall performance. A multistep approach is used to produce a balanced dataset in order to gain a better grasp of the aim as a baseline. Some pre-processing techniques are used on this dataset before it is fitted to any classifier. Missing values removal, stop-words removal, irrelevant attribute removal, and unnecessary space removal are some of the pre-processing strategies. This prepares the dataset for categorical encoding, which will be used to generate a feature vector.

1.3 Project Goals

The research proposes an automated solution based on machine learning-based classification approaches to prevent fraudulent job postings on the internet. Many classifiers are utilized for checking fraudulent posts on the web, and the results of those classifiers are compared in order to determine the optimum employment scam detection model. It aids in the detection of bogus job postings among a large number of candidates.

1.4 Summary

This chapter gave us the insight of the materials and tools that we have in the proposed system.

CHAPTER 2:
Methodology

2.1 Introduction

In this chapter we discuss the motivation due to which we thought of implementing this project. We will also discuss in this chapter as to why we have chosen the fake job posting detection field apart from all other fields to work it.

2.2 Motivation Towards Our Project

Our dream is to build a system where we can detect fake job posting more accurately than any other previous systems. Our system will help you to understand the online fraud and will help to stop applying in these fraud scams. We have developed a system which will be efficient, low cost, user-friendly & reliable. “Fake Job Posting Detection System” will detect the fake job posting and people won’t get deceived anymore by the scammers.

2.3 Summary

This chapter provided the idea about the motivation towards our project which aims to systematically automate the entire work structure of a diagnostic center.

CHAPTER 3: Related Work

3.1 Introduction

In this chapter we discuss the types of fake job posting detection that currently exist in the market. We also focus on the problems that the current system has and a proper justification will be provided as to why our system is the ideal one in the current circumstances.

3.2 Systems related to our project

There have been several articles about the detection of fake jobs and also related topics such as the identification of fake news, jobs and phishing emails. Some popular techniques are employed in a wide range of areas. Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna [6] conducted a paper about detecting spammers on social networks. They analyzed how spammers target social networking sites and how they operate. Nidhi Goyal, Niharika Sachdeva, and Ponnurangam Kumaraguru [7] performed some machine learning classifier for fraudulent job detection in recruitment. For this, they used a dataset, which had 157880 job postings data, and they got an f1-score of 96%. Zaharije Radivojevic and Branislava Cvijetic [8] conducted a paper about jobs advertised. They used the support vector machine algorithm and multinomial naive bayes to detect the job advertised. They achieved 90% accuracy. Joana Urbano [9] used machine learning to detect job posting inconsistencies and also use natural language processing. C. Jagadeesh [10] conducted a paper for job recruitment detection. In this experiment, two classifiers were used to detect scams. They used several machine learning algorithms and achieved 98.27% accuracy. Charan Lokku and Santhosh Puganuru [11] performed an experiment for the classification of gentility in job postings using a natural language processing model and they got the highest accuracy with the random forest algorithm. Shawni Dutta and Prof. Samir Kumar Bandyopadhyay [12] conducted research to determine the fake job. The major contribution of their research was to detect fake and real jobs using some machine learning methods. They tested various machine learning methods for training purposes. They used seven machine learning methods to detect fake job recruitment. They obtained their dataset from kaggle and trained their dataset with machine learning methods. They used single classifiers and ensemble classifiers for the detection of fake job requirements. They achieved an accuracy of 72.06% in the naive Bayesian classifier, 96.14% in the multi-layer perceptron classifier, 95.95% in the K-nearest

Neighbor Classifier, 97.2% in the Decision Tree Classifier, 98.27% in the Random Forest Classifier (RF), 97.46% in the Adaboost Classifier, and 97.65% in the gradient Boosting classifier. In their project, the best classifier was the ensemble classifier, which was the random forest classifier. Karri Sai Suresh Reddy and Karri Laskshman Reddy [13] conducted research on fake job recruitment detection in 2021. In their research, they used a dataset that was obtained from Kaggle. In their paper, they use a classifier for the detection of fake recruitment. They used single classifiers and ensemble classifiers. They used a random forest classifier to detect fake job recruitment. In their study, they achieved 98.27% accuracy in the random forest classifier. Anita, Nagarahan, Aditya Sairam, Ganesh, and Deepkumar [14] performed and analyzed some machine learning and deep learning techniques. In their research, they used three machine learning algorithms: The K-Nearest Neibor algorithm, the Random Forest algorithm, and one deep learning algorithm, which was the Bi-direction LSTM algorithm for the detection of fake jobs. When they applied machine learning algorithms, they achieved 97% accuracy in the K-nearest neighbor algorithm and 98% accuracy in the Random Forest algorithm. They got 98% accuracy in the Bi-LSTM neural network. Syed Mahbub and Eric Pardede [15] conducted research about online recruitment fraud detection using contextual features. They used the dataset which was obtained from Kaggle. For the detection of online recruitment fraud, they performed three algorithms: Naive Bayes (NB), J48, and JRip. After performing their algorithms, they achieved 94.29% accuracy in naive bayes, 96.19% in J48, and 83.42% in JRip. S. Vidros and Akoglu [16] in 2017 conducted an article which was an automatic detection of online recruitment fraud. They used a dataset that was obtained from kaggle. For this study, they used many machine learning algorithms. They achieved 50% accuracy in ZeroR, 72.22% accuracy in Logistic regression, 77.33% in OneR, 84.778% accuracy in J48, and 91.22% in random forest. Their best performing algorithm was the random forest. After review some research we know that the highest accuracy is 98.27% in random forest classifier.

3.3 Problems with the current systems

The problem with the current systems are that the prediction of these system's on fake job posting are not satisfactory. They are unable to get satisfactory accuracy. In order to stop fake job posting scams they need to impove their accuracy.

3.4 Proposed Solution

Our applied models logistic regression, random forest, support vector machine and extreme gradient boost. We applied those machine learning approaches because of better result. We review some paper and in our knowledge after reviewing the highest accuracy is 98.27 in random forest classifier and we want to get the highest accuracy with our proposed model.

3.5 Summary

The objective of this study is to conduct a comparative analysis of the prediction of fake job posting using machine learning approaches. The bulk of investigations were completed with an accuracy rate of roughly 96.1 %, which was regarded as outstanding. The novelty of this study is that we applied machine learning and natural processing algorithms and reached an accuracy of 96.1 percent, which is greater than in prior publications. The main contribution of our study is that we used a variety of well-known machine learning techniques to get our results. Many model comparisons have demonstrated their resilience, and the strategy may be formed from the study's research findings. Therefore, this chapter gives the idea about the current systems that are available and about the motivation towards our development.

CHAPTER 4:
TECHNICAL DESIGN

4.1 Introduction

In this chapter we discuss the aspect of the technical design of our system. By going through the system level design, it would be easier to conceptualize the entire data flow of the system.

4.2 Technical Design: Block diagram of system

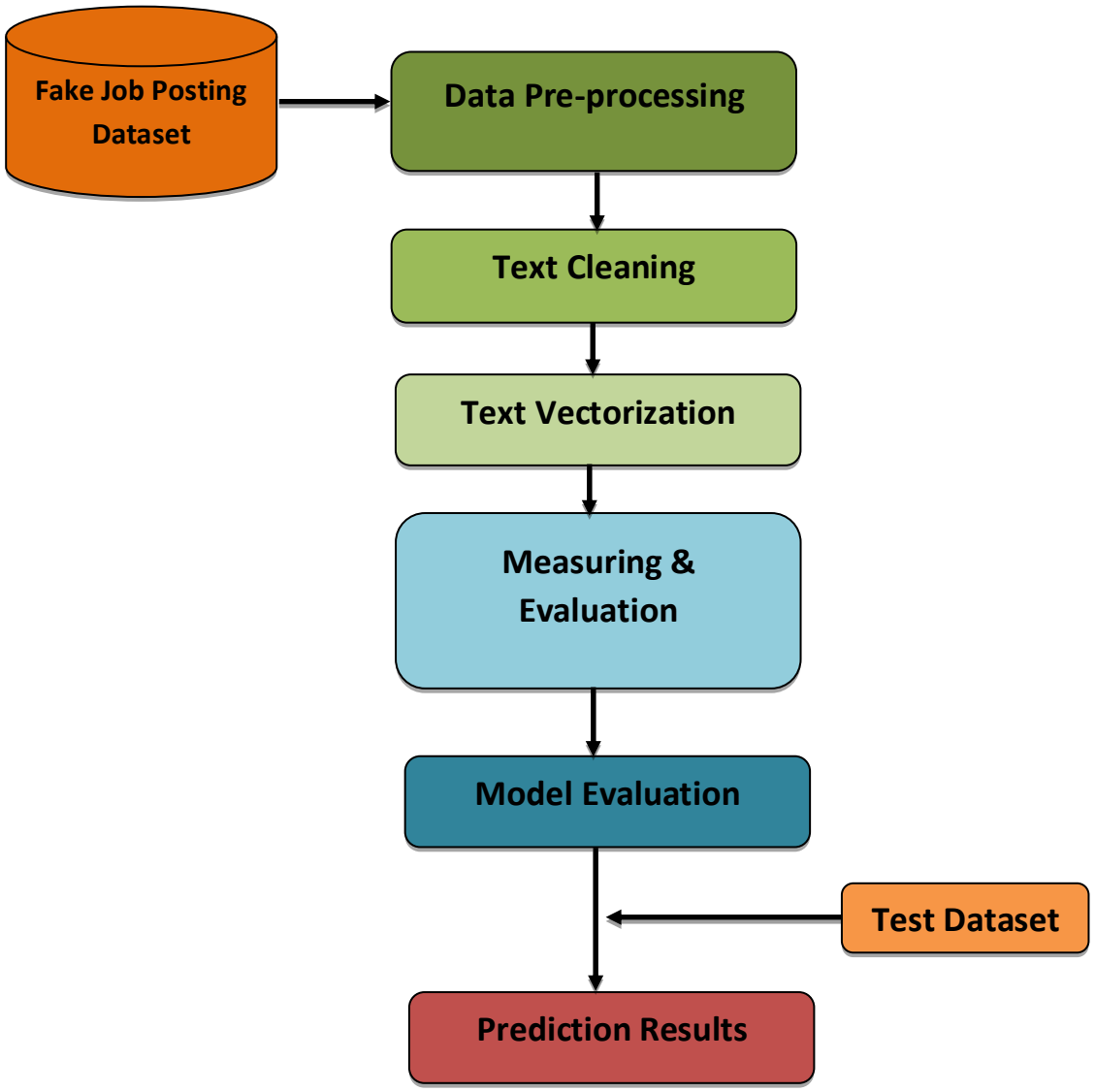


Figure 4.2.1: Block diagram of system

In figure 4.1, The block diagram has four main steps dataset, data preprocessing, data vectorization and model evaluation. The dataset used in this study is titled as “Real/Fake Job Posting Prediction” developed by Shivam Bansal in 2020. Dataset contains 17880 real and fake job posting data. Dataset is organized in 18 columns with textual and numeric data. Fake job posting must be cleaned during the preprocessing step: during this phase, we apply several cleaning and filtering techniques and filtering stop words on this dataset. Transformation of text to digital vectors is performed using the bag-of-words methodology with the TF-IDF method for computing the score of each word because most machine learning algorithms do not take text directly but digital vectors. Based on the results, we chose the most efficient classification method and then developed our classification model. Training and evaluation of the classification model using performance metrics (confusion matrix, classification rate), as well as testing the model on a collection of test data that represents a set of unclassified false or genuine job advertisements in order to predict fake job listings. We generated two matrices with the aim of evaluating performance.

4.3 Summary

As mentioned earlier, the technical design has enabled us to get a clear picture of how our system is operating. Therefore, considering the above data flow diagram we can comprehend the method in which our system is being operated.

CHAPTER 5: Data Analysis

5.1 Introduction

In this chapter we discuss the data analysis, data cleaning and preprocessing.

5.2 Data Analysis:

Data analysis is the process of cleaning, manipulating, and modeling data to extract useable information. The purpose of data analysis is to extract useful information from data and make decisions based on it. When we make a decision in our daily lives, we examine what occurred the last time we made that decision or what would happen if we made that decision. This is a fundamental form of data analysis. Looking backwards or forwards in time and making judgments based on our discoveries is all that this entails. We do this by reminiscing about the past or daydreaming about the future. So that's all there is to it when it comes to data analysis. For business purposes, an analyst today does data analysis.

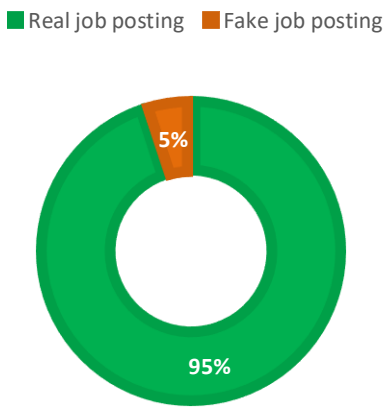


Figure 5.1: Percentage of real and fake data

After the data has been obtained, data analysis must be performed in order to get insight into the data. Pandas, NumPy, matplotlib, and seaborn are Python modules and frameworks that let us visualize the distribution of data and offer a rudimentary understanding of real and fake jobs. The analysis process gives us an idea of how unclean our data is, and hence data cleaning is required. Figure 2 shows the percentage of real and fake jobs in the dataset, where 95% of the data is about real jobs and 5% of the data is about fake jobs. Figure 3 shows the

number of fake jobs and real jobs in the dataset where X-axis is the classification of fake and real jobs and Y-axis is the number of fake and real jobs.

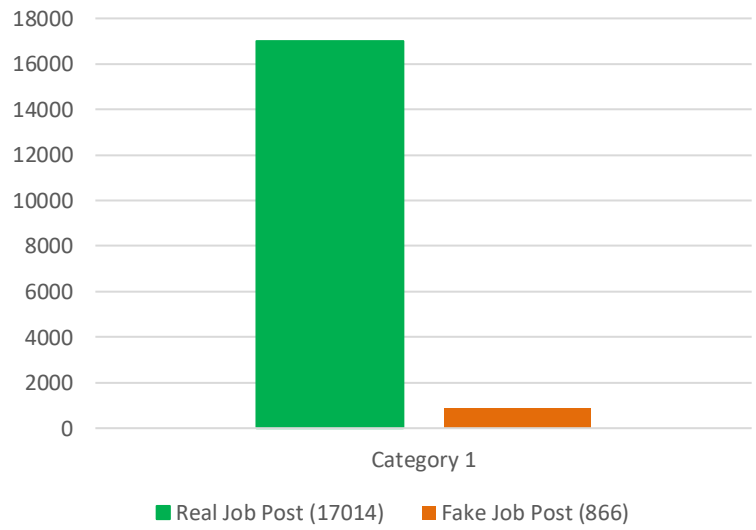


Figure 5.2: Graph of real and fake data

Collecting the raw data sets, we'll need to answer the question we've selected is the first stage in the data analysis process. Cleaning the data so that it can be made ready for analysis. This includes removing duplicate and anomalous data, addressing inconsistencies, standardizing data structure and format, and dealing with white spaces and other grammatical issues, to name a few. Analyze the facts you've gathered. By changing the data using various data analysis techniques and tools, you may begin to identify trends, correlations, outliers, and changes. You may use data mining to uncover trends in databases or data visualization tools to convert data into a graphical format that's easier to understand during this phase.

5.2.1 Data Cleaning and Preprocessing

We notice numerous null values and some textual data that needs to be purged after performing an analysis of the data on the provided data. As a result, we first look at all of the null values in each column first, and then we delete the columns with a large number of null values. After this, we check the unnecessary words, which do not contribute to the detection of fake jobs. Following the removal of extraneous words from the data, all textual data is

consolidated into a single column that machine learning techniques may use. Overall, Python libraries like pandas and Natural Language Programming (NLP) packages like textblob have been used to clean input.

Data preparation is required to use data mining methods because raw data must be transformed through a well-set of data. Whenever it comes to actual data, it's common for information to be inaccurate and poorly formatted. The success of any data analytics project is closely linked to the role of data analysis. Data validation and data imputation are both part of the preprocessing phase. The purpose of data validation is to ensure that the data is accurate and comprehensive. Information restoration can be done manually or automatically using BPA programming to rectify errors and fill in voids. Data preparation is used in both database-driven and rules-based systems. Data preparation is crucial in machine learning (ML) processes to ensure that huge amounts of information are prepared in a way that enables machine learning techniques to handle and assess the information they include. During preprocessing, the data goes through multiple steps:

Data cleansing might include replacing null values or removing rows containing incomplete information, flattening data redundancy, and correcting errors. Because computers can't use data they don't understand, smoothing noisy data is critical for ML datasets. Binning data into equal-sized pieces, matching this to a straight or multivariable function, or grouping it into clusters of comparable data are all ways to clean data (clustering). Mishaps can lead to data discrepancies (the information was stored in the wrong field). To avoid providing that data item an advantage, duplicate values should be removed via reduction (bias). The process of merging data from various formats and resolving data conflicts is known as data integration. Data Standardization and Generalization: The data has been validated and modified. The data transformation guarantees that no information is replicated; everything is kept in one place, so all interactions are acceptable. Databases may become slower, more expensive to access, and more difficult to store as data amounts expand. In a data warehouse, data reduction seeks to produce a cleaner piece of information. Analyzing may be

accomplished in a variety of ways. Everything below a certain degree of importance, for example, is deleted once a subset of key attributes has been picked for their relevance. To reduce the bulk of the data, encoding strategies might be used. If all of the original data can be retrieved after compression, the approach is considered lossless. A lousy reduction is one that results in the loss of some data. Grouping can then be used to cut down the number of data items by combining many operations into a single weekly or monthly value. If the information is factorized, raw values can be substituted with interval levels. To decrease the number of values for continuous attributes, this phase includes splitting the range of attribute intervals. Due to time, storage, or memory constraints, a dataset may be too vast or complicated to handle. Utilizing sampling techniques, only a portion of the dataset can be chosen and dealt with, as long as it has similar features to the original. Cleaning, manipulating, and modeling data to obtain valuable information is part of the data analysis approach. The purpose of data analysis is to extract useful information from data and make decisions based on it. A dataset is also too large as well as difficult to process due to scheduling, storage, or memory restrictions. Just a portion of the dataset can be picked and dealt with using data collection techniques as long as it has similar qualities to the original. Cleaning, manipulating, and modeling data to extract useable information is the process of data analysis. Data analysis is used to extract relevant information from data and make decisions based on that knowledge.

5.3 Summary

This chapter provided all the necessary details on the data analysis, data cleaning and preprocessing.

CHAPTER 6:
Algorithms

6.1 Introduction

In this chapter we explore the different kinds of algorithms that we have used in our project.

6.2 Algorithms

We will use natural programming language and machine learning algorithm to identify fake job advertisement one the data examine is complete. We will apply three machine learning algorithms to identify fake job advertisements. Those classifiers are: Random Forest classifier, Logistic regression classifier, Support vector machine classifier and Extreme Gradient Boost Classifier.

6.2.1 Logistic regression Classifier

Logistic regression classifier is expressed by equation, much like linear regression. The input data (x) is linearly blended with factor values to observe an output (y). The final value is a binary value (0 or 1) rather than a numerical number, which is a key difference from linear regression.[21]

$$Y = \frac{e^{b_0 + b_1 * x}}{1 + e^{b_0 + b_1 * x}} \qquad (1)$$

Where y showed the result, b0 appears to be anticipated term, and input data value's b1 coefficient (x). Using our training data, we must determine the b coefficient for each column in our input data. The values (or b's) in the formula represent the genuine reflection of the system, which we may store or record in a file.

Figure 6.2.1.1 demonstrates a logistic curve with only 0 to 1 values. In logistic regression, the natural logarithm "odds" something like the goal property, more than the probability, is used to construct the curve, which is comparable to linear regression. Additionally, predictors do not need to be distributed [22] randomly or have the same variability in each category. The variable (b0) pushes the line left and right in logistic regression, whereas the slope (b1) determines how steep the curve is.

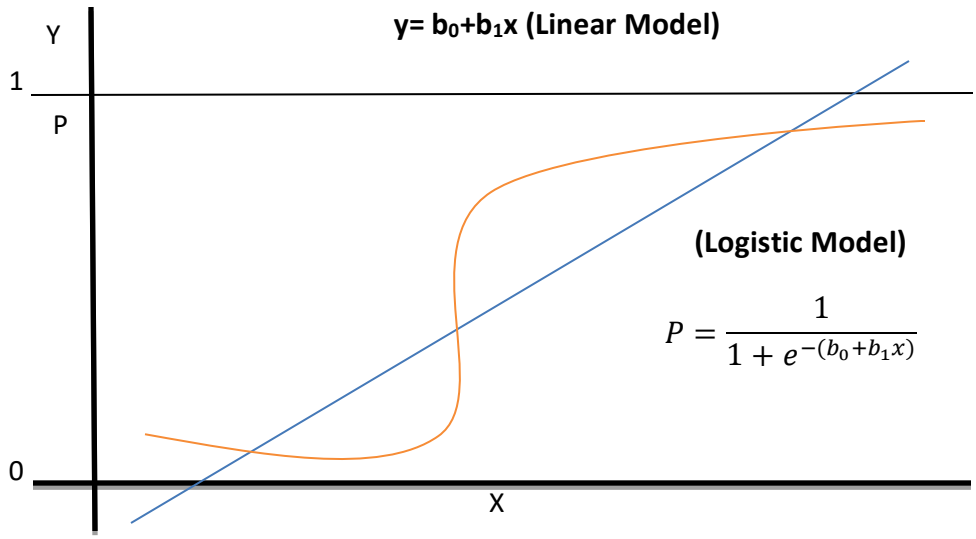


Figure 6.2.1.1: Logistic curve in logistic regression

6.2.2 Random Forest Classifier

The random forest seems to be a machine learning approach that may be used to solve classification and regression issues. It makes use of ensemble approaches, which constitute a method of addressing difficult problems by combining numerous classifiers. The random forest strategy is composed of a number of decision trees. The "forest" of a random forest technique is trained via skimming or stochastic aggregation. Bagging is a meta-algorithm that combines machine learning methods to increase accuracy. The outcome is determined by the random forest technique, which is based on decision tree forecasts. It predicts by merging and otherwise averaging the findings of many trees. Like the quantity of trees increases, so does the accuracy of the output. The decision tree classifier's shortcomings are solved by the random forest technique. [23] It improves accuracy while reducing overfitting in datasets. It can produce projections without a large number of packaging configurations (like scikit-learn). The nodes throughout the decision tree indicate the qualities used to forecast the outcome. Those selection nodes are linked to the leaves. The 3 kinds of branches present in a decision tree are depicted in the diagram below.

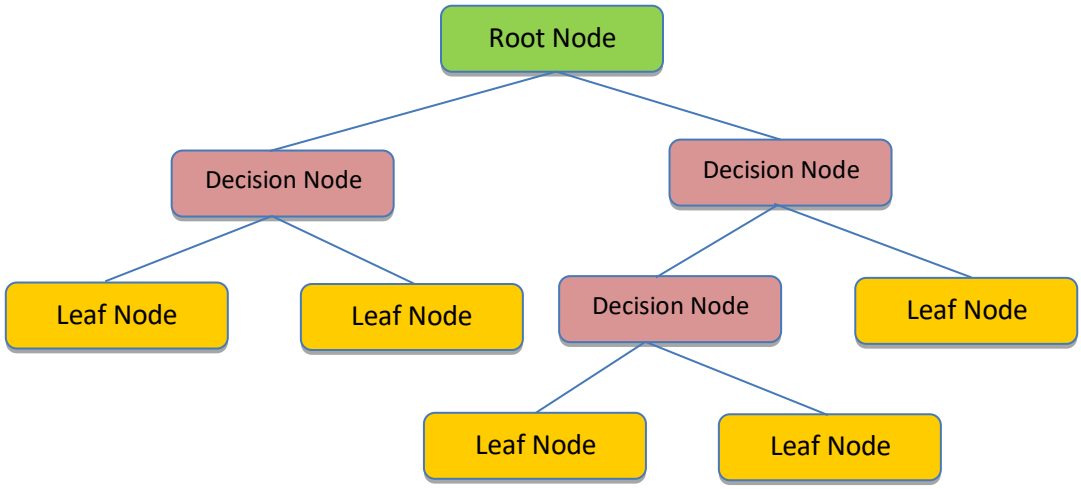


Figure 6.2.2.1 : Three kinds of nodes inside a decision tree [24].

Figure 6.2.2.1 shows three different sorts of nodes in a decision tree. Information theory can provide more details according to how decision trees work. Entropy and information gain are the two most important components of decision trees. The entropy metric is used to determine uncertainty. Information strength is a measure of how much ambiguity is eliminated in the target feature given a set of independent factors. The information gathered is utilized to train decision trees. It aids in the reduction of ambiguity in these trees. A great deal of ambiguity (a variable that impacts) has been reduced as a result of large information gain. The utilization of entropy and collective progress is required when splitting branches, which is a core fundamental in the creation of decision trees.

6.2.3 Extreme Gradient Boost Classifier Algorithm

The Extreme Gradient Boost, abbreviated as XGBoost, is a machine learning approach used to create gradient-boosted trees. What are decision trees used for? When all of this comes to evaluating unorganized data such as images, unstructured text data, and so on, ANN models are frequently at the top of the list. However, when this refers to organized or moderately organized data, decision trees are without a doubt the best alternative. XGBoost was designed to drastically realize the full potential and reliability of ML algorithms and has done so excellently. XGBoost is a gradient boosting decision tree application with great speed and performance. Because it is mainly concerned with processing speed and model performance, the library offers little frills. [25] It does, however, provide a comprehensive set of advanced functions. XGBoost can do parallel processing on a single machine because it contains both a tree new approach and a strategies supplied instructional strategy. It really makes it 5 times higher efficient than in any other gradient boosting approach now in use.

6.2.4 Support Vector Machine Classifier

A Support Vector Machine Classifier is a popular supervised teaching method for classification or regression. However, there is a large use of it in machine learning to solve classification difficulties. The goal of the SVM approach is to discover the best line or aim component for dividing an n-dimensional region into divisions such that further data points may be readily added in the future. A hyperplane is a boundary that is the best choice. [26] SVM is used to choose the greatest number of points or vectors that will aid in the creation of the hyperplane. Support vectors are extreme examples, and the approach is known as a "support vector machine."

6.3 Summary

These are the different kinds of algorithms that we have used in our project.

CHAPTER 7: SKILLS

7.1 Introduction

In this chapter we discuss the skills that we have obtained in order to develop this massive sophisticated system.

7.2 Skills obtained

Through this project the following skills have been developed:

7.2.1 Skill in Programming & Tools

7.2.1.1 Anaconda (Python distribution)

Anaconda is a data processing and scientific computing platform based on Python. It includes a number of third-party libraries that are really useful. Installing Anaconda is the same as installing Python and some frequently used libraries like Numpy, Pandas, Scrip, and Matplotlib, and it makes the process considerably easier than installing standard Python. If you don't use Anaconda and instead use python.org to install Python, you'll need to use pip to install the relevant libraries one by one. Because it is inconvenient.

Our experiences with it on both Windows and Linux have been quite favorable. It's fairly comprehensive, and it avoids issues with creating libraries from source code, which afflict one-by-one installations of such libraries using tools like pip.

Starting with 3.5 or 3.6 is a good idea because 2.7 is nearing the end of its lifecycle, even though many apps still rely on it.

In terms of tutorials, Python's own documentation is excellent for learning the language.

7.2.1.2 Jupyter Notebook

The Jupyter Notebook App is a web-based server-client tool for editing and running notebook papers. As explained in this paper, the Jupyter Notebook App can be run locally on a computer without internet access (as explained in this paper) or remotely on a server and accessible via the internet.

The Jupyter Notebook App contains a "Dashboard" (Notebook Dashboard) and a "control panel" that shows local files and allows you to open notebook documents or shut down their kernels, in addition to displaying, editing, and running notebook documents.

7.2.1.3 Colaboratory or Colab

Google Research's Colaboratory, or "Colab" for short, is a product. Colab is a web-based Python editor that allows anyone to write and run arbitrary Python code. It's notably useful for machine learning, data analysis, and education. Colab is a hosted Jupyter notebook service that doesn't require any setup and offers free access to computational resources, including GPUs.

7.2.2 Skill in Machine Learning Algorithms

7.2.2.1 Logistic regression Classifier

Logistic regression classifier is expressed by equation, much like linear regression. The input data (x) is linearly blended with factor values to observe an output (y). The final value is a binary value (0 or 1) rather than a numerical number, which is a key difference from linear regression Where y showed the result, b0 appears to be anticipated term, and input data value's b1 coefficient (x). Using our training data, we must determine the b coefficient for each column in our input data. The values (or b's) in the formula represent the genuine reflection of the system, which we may store or record in a file. In logistic regression, the natural logarithm "odds" something like the goal property, more than the probability, is used to construct the curve, which is comparable to linear regression. Additionally, predictors do not need to be distributed randomly or have the same variability in each category. The variable (b0) pushes the line left and right in logistic regression, whereas the slope (b1) determines how steep the curve is.

7.2.2.2 Random Forest Classifier

The random forest seems to be a machine learning approach that may be used to solve classification and regression issues. It makes use of ensemble approaches, which constitute a method of addressing difficult problems by combining numerous classifiers. The random forest strategy is composed of a number of decision trees. The "forest" of a random forest technique is trained via skimming or stochastic aggregation. Bagging is a meta-algorithm that combines machine learning methods to increase accuracy. The outcome is determined by the random forest technique, which is based on decision tree forecasts. It predicts by merging and otherwise averaging the findings of many trees. Like the quantity of trees increases, so does the accuracy of the output. The decision tree classifier's shortcomings are solved by the

random forest technique. It improves accuracy while reducing overfitting in datasets. It can produce projections without a large number of packaging configurations (like scikit-learn). The nodes throughout the decision tree indicate the qualities used to forecast the outcome. Those selection nodes are linked to the leaves. The 3 kinds of branches present in a decision tree are depicted in the diagram below. Information theory can provide more details according to how decision trees work. Entropy and information gain are the two most important components of decision trees. The entropy metric is used to determine uncertainty. Information strength is a measure of how much ambiguity is eliminated in the target feature given a set of independent factors. The information gathered is utilized to train decision trees. It aids in the reduction of ambiguity in these trees. A great deal of ambiguity (a variable that impacts) has been reduced as a result of large information gain. The utilization of entropy and collective progress is required when splitting branches, which is a core fundamental in the creation of decision trees.

7.3 Summary

In this chapter we discussed the list of skills that have been obtained throughout the process of developing and materializing this system.

CHAPTER 8: ESSENTIAL PARTS AND DEVICES

8.1 Introduction

In this chapter, we shed light on the tools that we used to develop this sophisticated system and we also discuss what tools will be required if one wants to test this system in there one spheres.

8.2 Design Requirements

We used the following things for our system:

- Anaconda
- Jupyter Notebook
- Libraries in notebook

8.2.1 Software:

- Anaconda

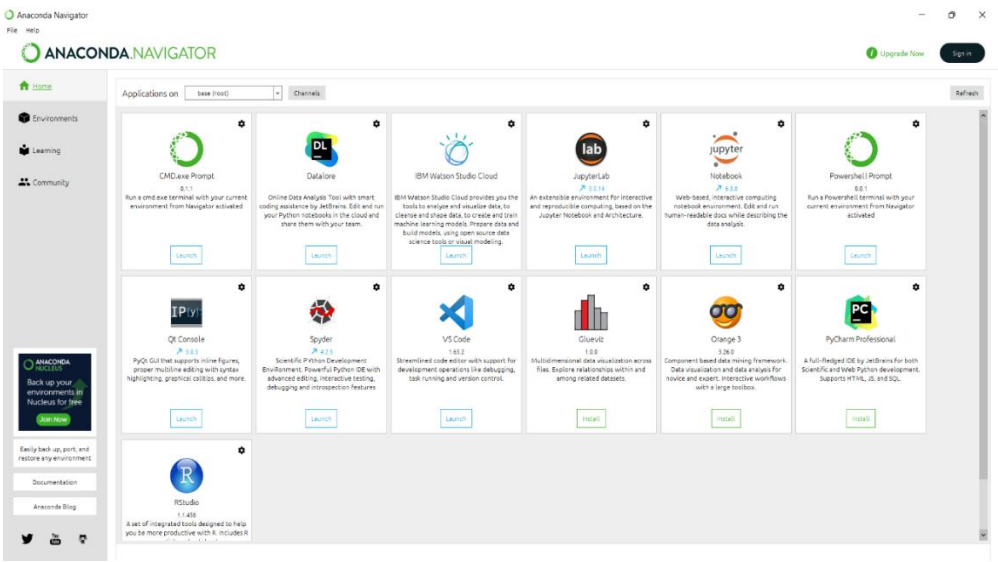


Figure 8.1: Anaconda navigator

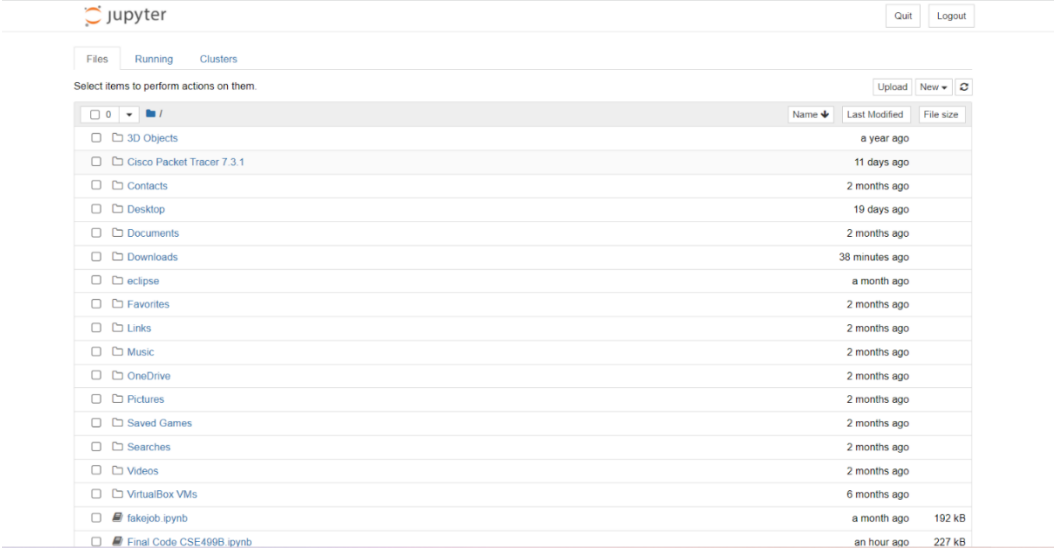


Figure 8.2: Jupyter Notebook

8.3 Operating System:

- Win-XP, Win-7, Win-8 or higher version
- Linux or any other higher version.

CHAPTER 9: Working Sheets

9.1 Introduction:

In this chapter, we will observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase. We shall also see the financial foundation of this project and furthermore the feasibility study should be also discussed.

9.2 Work Breakdown Structure

In order to develop this system, we gave importance to scheduling because we believed if we want to provide the best of quality then we must give due importance to scheduling which helped us to garner better results. The figure below focuses work we had accomplished.

Grant Chart Showing expected timeline of progress

Working Details	**Month-1**	**Month-2**	**Month-3**	**Month-4**	**Month-5**	**Month-6**	**Month-7**
Related Research Study							
Collecting informations							
Analyze Dataset collections							
Model Selection							
Training and testing data							
Comparing with other model.							
Prediction and accuracy							
Feedback and update							
Final Report							

9.2.1 Work Planning

1. Find out related research and journal papers and go through them. We will select papers based on related categories.
2. Comparing those research papers and we will find out the alternative best solutions which will be more efficient.
3. Find out the best datasets for our project which dataset will have our required features and latest information based.
4. We will analyze the features of the dataset and select the best model for the dataset. We will train our model with the trained dataset and test the model with the test dataset.
5. We will use another model with the same dataset and compare the accuracy with our previous selected model accuracy.
6. Finally, we will find the model with best accuracy and finalize the model to provide in the prediction of fake job posting.
7. We will collect the feedback and update our model if required.
8. We will wrap up our final report and complete our paper.

Budget

SL #	Item Description	Amount
1	Study and research on	6000tk
2	Computers, Laptops, Smartphones.	120000tk
3	Survey cost	2000tk
4	Advertisement through Facebook, youtube, etc	10000tk
	Total	138000 tk

CHAPTER 10: FUTURE WORK

10.1 Introduction

This chapter discusses the future scope or the implementation of this system.

10.2 Future Scope of Work

The main objective of developing this system is to help people so that they don't get scammed. The system can be more improved than the current form. We can test alternative classical classifiers with different features and deep learning models with various word representations, as well as combine the two methods. We will compare classical machine learning with deep learning on this dataset in order to obtain a thorough picture of this research. Besides, we intend to use and evaluate our approach for hierarchy-based and path-based fact-checking algorithms

10.3 Summary

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed system. The system may need some research for different applications, though the principle of the designed system will remain as it is.

CHAPTER 11: DESIGN IMPACT

11.1 Introduction

In this chapter, we discuss about the various impacts that our system has been able to generate.

11.2 Environmental Impact

By introducing this system, we can reduce scammers, when our system will be able to detect scam post and people will recognize fake post then they will not apply and the scammers will be discouraged to create such fake post in future.

11.3 Economic Impact

The economic impact that this system entails is that by introducing this system a huge number of people won't get scammed by the scammers and they won't lose their money to the scammers.

11.4 Social Impact

The Fake Job Posting Detection system will be socially acceptable as this kind of system is the need of the hour. In this era of ours, an automated system for fake job detection is very necessary that can give very much accurate results. Therefore, our system is no exception.

11.5 Sustainability

Our system has been able to deal with huge number of fake jobs posting'. When the numbers of tests are conducted simultaneously our system remains stable. Therefore based upon these facts and continuous testing, our system is sustainable.

11.6 Summary

This chapter has covered the different types of impacts that our system offers and those has been described and discussed. From the above given impacts we can conclude that our designed system is good enough to use under any circumstance.

CHAPTER 12: RESULTS

12.1 Introduction

This chapter of the report contains the results that we achieved throughout the course of using this system. The identification of fake job postings will direct job searchers only to real employment offers from firms. Three machine learning methods are offered as countermeasures in this study to combat fake job posting detection. A supervised technique is employed to demonstrate the usage of many classifiers for fake job posting detection. Once machine learning and natural language processing (NLP) algorithms have been applied, we must analyze and compare them to select the optimum model for the categorization of false jobs from a pool of job postings. Our suggested method attained a precision of 96.1% in the multinomial naive bayes algorithm and 96.00% in the passive aggressive algorithm. 96.1% is significantly greater than current approaches.

12.2 Results Achieved

12.2.1 Confusion Matrix

A confusion matrix is a N x N matrix used to assess the effectiveness of a classification model, where N is the number of target classes. The matrix compares the actual target values to the predictions of the machine learning algorithms. This gives us a detailed view of how well our classification model is performing and the sorts of errors it produces.

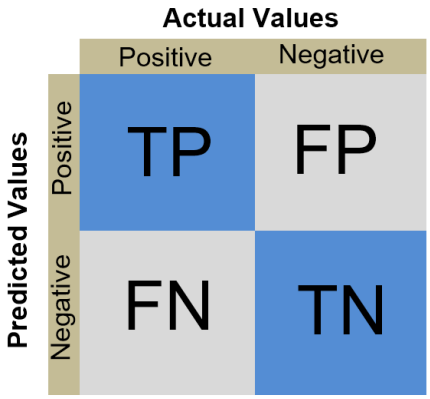


Figure 12.2.1.1: Sample of Confusion matrix.

Figure 12.2.1.1 demonstrates that "True Positive" signifies that the projected value reflects the real value, and that the real value was positive, just as the model anticipated. The True Negative is the estimated value that corresponds to the actual value, and the model anticipated a negative number since the actual value was negative. A "False Positive" occurs when the expected value is wrongly projected (FP). Despite the fact that the actual number was negative, the model predicted that it would be positive. The False Negative (FN) is the value that was mistakenly predicted. The model predicted a negative outcome, while the actual result was positive.

The system created a confusion matrix with fake and real job post values in the columns and predicted values in the rows. A confusion matrix would be a summary of the outcomes of a machine learning model's predictions. True and false predictions are totaled and broken down by class in the confusion matrix, and the false positive and negative, as well as true positive and negative matrices, are generated using the following formulae for n x m matrices. [30] Figure 12.2.1.2 represents the confusion matrix of the logistic regression classifier algorithm. This confusion matrix shows that our model predicts 5131 fake job postings out of 5134 fake job data points, and 3 out of 5134 data points are fake job postings, but our model predicts those as real. On the other hand, our model predicts true negative values of 172 out of 230 data points and false negative values of 58 out of 230 data points, which are real jobs but predicted as fake jobs. With this confusion matrix, we got the classification report.

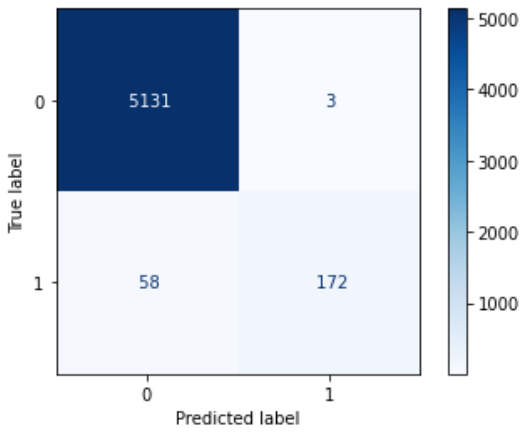


Figure 12.2.1.2: Confusion matrix of logistic regression classifier algorithm

Figure 12.2.1.3 represents the confusion matrix of the random forest classifier algorithm. This confusion matrix shows that our model predicts 5133 fake job postings out of 5134 fake job data points, and 1 out of 5134 data points are fake job postings, but our model predicts those as real. On the other hand, our model predicts true negative values of 153 out of 230 data points and false negative values of 77 out of 230 data points, which are real jobs but predicted as fake jobs. With this confusion matrix, we got the classification report.

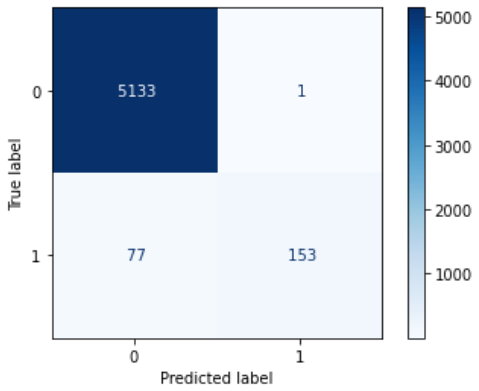


Figure 12.2.1.3: Confusion matrix of random forest classifier algorithm

12.2.2 Model Evaluation

The models' precision, F-score, accuracy, and recall are used to evaluate performance. The performance of the proposed model was assessed with regard to the number of true and false positive values. There are two kinds of negatives: real negatives and false negatives. Recall, also described simply as sensitivity, is the ratio at which the affected photographs are accurately identified among all the data. Precision is the inverse of recall. The F1-score is an accuracy and recall statistic that indicates how frequently the predicted value is right. It is also known as the symmetrical mean of p and r in mathematics. These are the equations shown below. Matrices may be used to assess the performance of a system both before and after the model has been built. Accuracy is a measurement of how precise something is. The following equations provide the mathematical formulae for determining accuracy [31]:

$$accuracy = \frac{TN + TP}{TP + FP + FN + TN} \quad (7)$$

$$accuracy = \frac{correct\ prediction}{total\ number\ of\ examples} \quad (8)$$

Recall, usually referred to as sensitivity, seems to be the rate at which a true value is properly identified from a set of values. To calculate recall in an equation, use the following expression (9). Accuracy in recognition is defined as precision. The number of times the model's positive prediction was right may be calculated using the mathematical method below, which is very closely related to the model's high identification in equation (10).

$$recall = \frac{TP}{TP + FN} \quad (9)$$

$$precision = \frac{TP}{TP + FP} \quad (10)$$

The F1-score is a unique matrix that evaluates both precision and recall and may be used to quantify the performance of a classifier for both recall and precision. It's also known as the harmonically precise technique of precision and recall in mathematics. The F1-score is calculated using the following equation:

$$F_1 score = \frac{2\,TP}{2\,TP + FP + FN} \quad (11)$$

$$F_1 score = \frac{2\,pr}{p + r} \quad (12)$$

The classification analysis for the logistic regression algorithm is shown in Figure 12.2.2.1 In this study, the logistic regression method has an ultimate F1-score of 99 percent. Individuals have an F1-score of 99% for phony job postings and an F1-score of 85% for legitimate job postings. Figure 10 also depicts the precision and recall value. Even after fine-tuning, the accuracy remained constant.

Classification report of Logistic Regression

	precision	recall	f1-score	support
0	0.99	1.00	0.99	5134
1	0.98	0.75	0.85	230
accuracy			0.99	5364
macro avg	0.99	0.87	0.92	5364
weighted avg	0.99	0.99	0.99	5364

Figure 12.2.2.1: Logistic regression classifier classification report.

The classification analysis for the random forest algorithm is shown in Figure 12.2.2.2. In this study, the random forest method has an ultimate F1-score of 99 percent. Individuals have an F1-score of 99% for phony job postings and an F1-score of 79% for legitimate job postings. Figure 11 also depicts the precision and recall value. Even after fine-tuning, the accuracy remained constant.

Classification report of Random Forest Classifier

	precision	recall	f1-score	support
0	0.98	1.00	0.99	5134
1	0.99	0.66	0.79	230
accuracy			0.99	5364
macro avg	0.99	0.83	0.89	5364
weighted avg	0.99	0.99	0.98	5364

Figure 12.2.2.2: Random Forest classifier classification report

The classification analysis for the support vector machine algorithm is shown in Figure 12.2.2.3. In this study, the support vector machine method has an ultimate F1-score of 99 percent. Individuals have an F1-score of 99% for phony job postings and an F1-score of 73% for legitimate job postings. Figure 12 also depicts the precision and recall value.

Model	State	Recall	Precision	F_1 score
Logistic Regression Classifier	Real	0.75	0.98	0.85
Logistic Regression Classifier	Fake	1.00	0.99	0.99
Random Forest Classifier	Real	0.66	0.98	0.79
Random Forest Classifier	Fake	1.00	0.99	0.99

Table 12.2.1: Model Evaluation

12.2.3 Model Comparison

In this study, we compared the four classifier algorithms. In this study, we used four machine learning approaches, which are logistic regression, random forest, support vector, and extreme gradient boost classifiers. Those classifiers gave us great accuracy. With that accuracy, we plotted a graph that shows the highest accuracy. Figure 12.2.3.1 shows the accuracy comparison of our used model. We got an accuracy of 98.86% in the logistic regression classifier, 98.54% in the random forest classifier, 98.19% in the support vector classifier, and 97.66% in the extreme boost classifier. We assume that the accuracy and f1-score of logistic regression are higher than those of any other classifier.

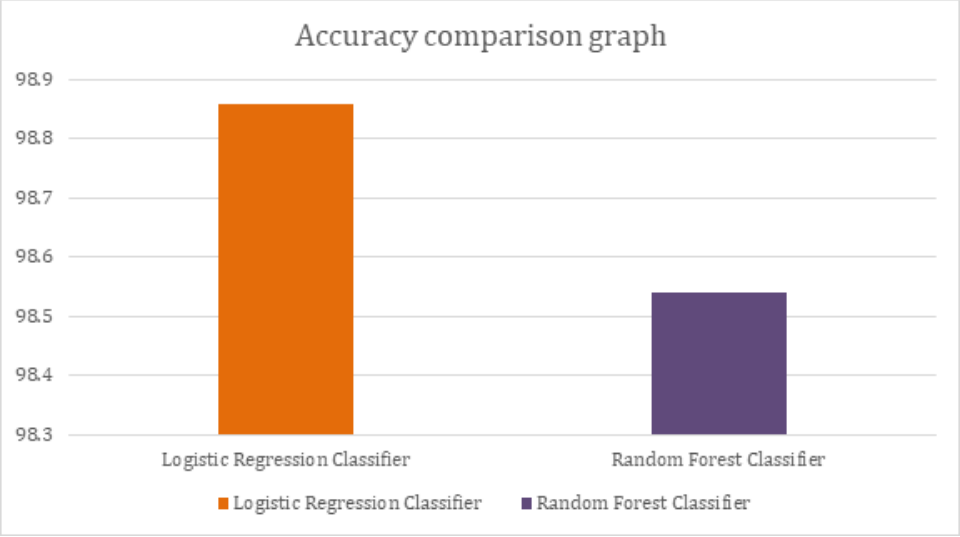


Figure 12.2.3.1: The graph of accuracy in this study.

This study's logistic regression, random forest, support vector, and extreme gradient boost classifiers were compared to the previous algorithms. logistic regression classifier topped the algorithms in the relevant publications in terms of performance, accuracy, and efficiency. In terms of accuracy, precision, and recall, the developed logistic regression approach topped those in previous studies. Table 12.2.3.1 provides extensive comparative analysis among many approaches.

Reference paper (models name)	Accuracy (%)	Model in this study	Accuracy (%)
Ref [8] Multinomial Naïve Bayes	90.00		
Ref [10] Random Forest Classifier	98.27		
Ref [12] Random Forest Classifier	98.27	Random Forest	98.54
Ref [13] Random Forest Classifier	98.27		
Ref [15] Naïve Bayes Classifier	96.19		
Ref [16] Logistic Regression	72.22	Logistic Regression	98.86

Table 12.2.3.1: Table of model comparison

12.3 Summary

This chapter has covered the different types of results that we have managed to obtain throughout the course.

CHAPTER 13: CONCLUSION

Conclusion:

In this study, Job searchers will be guided by fake job posting detection to only get authentic job offers from firms. In this study, numerous machine learning methods are offered as defenses for detecting fake job posting. The use of numerous classifiers for fake job detection is demonstrated using a supervised approach. The internet's emergence has tremendously eased the recruitment process. In recent years, scammers have appeared in the online recruiting market. Spammers use internet recruitment scams to fool individuals with enticing employment offers. Some of the advertised roles are really phantom positions created to steal critical data. The creation of an automated method for detecting fake job advertisements and alerting people so they don't apply has piqued people's curiosity. A machine learning technique use a range of categorization algorithms to detect fake jobs. Malicious user profiles and echo chamber effects are hallmarks of fake news on social media. The proposed technique achieved an accuracy of 96.1 percent in multinomial naive bayes, which is significantly higher than previous approaches.

Future work:

In the future, we'd like to increase the dataset's amount as well as its quality. On this corpus, we also want to test alternative classical classifiers with different features and deep learning models with various word representations, as well as combine the two methods. We will compare classical machine learning with deep learning on this dataset in order to obtain a thorough picture of this research. Besides, we intend to use and evaluate our approach for hierarchy-based and path-based fact-checking algorithms. In future studies, we want to evaluate alternative algorithms for learning heterogeneous documents like CVs in order to construct an integrated framework and examine user attributes.

BIBLIOGRAPHY

1. "15 Common Job Search Scams and How to Protect Yourself," Flexjobs, 2007. [Online]. Available: https://www.flexjobs.com/blog/post/common-job-search-scams-how-to-protect-yourself-v2/.
2. "What Is a Fake Job Posting and Ways to Spot it?," Omnes, 2020. [Online]. Available: https://www.omnesgroup.com/fake-job-posting/.
3. B. Alghamdi and F. Alharby, "An intelligent model for online recruitment fraud detection," *Scientific research*, vol. 10, no. 3, p. 22, 2019.
4. V. L. Rubin, N. Conroy, Y. Chen and S. Cornwell, "'Fake news or truth? using satirical cues to detect potentially misleading news," in *University of Western Ontario*, London, Ontario, CANADA, 2016.
5. H. Ahmed, I. Traore and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," *Springer link*, pp. 127-138, 2017.
6. G. Stringhini, C. Kruegel and G. Vigna, "Detecting spammers on social networks," *Association for computing machinery digital library*, pp. 1-9, 2010.
7. N. Goyal, N. Sachdeva and P. Kumaraguru, "Spy the Lie: Fraudulent Jobs Detection in Recruitment Domain using Knowledge Graphs," *Springer link*, pp. 612-623, 2021.
8. B. Cvijetic and Z. Radivojevi, "Application of Machine Learning in the Process of Classification of Advertised Jobs," *International journal of electrical engineering and computing*, vol. 4, no. 2, pp. 93-100, 2020.
9. U. Joana, M. Couto, G. Rocha and Henriq, "Inconsistency Detection in Job Postings.," in *In 3rd Conference on language, Data and knowledge*, 2021.
10. C. Jagadeesh, D. P. R. Kshirsagar, G. Sarayu, G. Gouthami and B. Manasa, "Artificial intelligence based Fake Job Recruitment Detection Using.," *Journal of Engineering Sciences*, 2021.
11. C. Lokku, K. N. S. Kolli and S. Puganuru, "Classification of Genuinity in Job Posting Using," *International journal for research*, vol. 9, 2021.
12. S. Dutta and S. K. Bandyopadhyay, "Fake Job Recruitment Detection Using Machine Learning Approach.," *International Journal of Engineering Trends and Technology*, vol. 68, no. 4, pp. 48-53, 2020.
13. K. S. S. Reddy and K. L. Reddy, "Fake Job Recruitment Detection," *Journal of Emerging Technologies and Innovative Research*, vol. 8, no. 8, 2021.
14. C. Anita, . P. Nagarajan , G. Sairam, P. Ganesh and G. D. Kumar, "Fake Job Detection and Analysis Using Machine Learning and Deep Learning Algorithms," *Revista Gente-Gestao Inovacao E Technologias*, vol. 11, no. 2, pp. 642-650, 2021.
15. S. Mahbub and E. Paedede, "Using Contextual Features for Online Recruitment Fraud Detection," *Ais Electronic Libray*, 2018.
16. G. Kambourakis, L. Akoglu, S. Vidros and C. Kolias, "Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset," *Mdpi open access journals*, vol. 9, no. 1, p. 6, 2017.
17. S. Bansal, "Real / Fake Job Posting Prediction," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-predictionl..
18. "Data Preprocessing," Techopedia, 2021. [Online]. Available: https://www.techopedia.com/definition/14650/data-preprocessing.
19. "What Is Data Preprocessing & What Are The Steps Involved?," Monkeylearn, [Online]. Available: https://monkeylearn.com/blog/data-preprocessing/.
20. "Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data," Tableau, [Online]. Available: https://www.tableau.com/learn/articles/what-is-data-cleaning.

21. "Logistic Regression — Detailed Overview," Towards science, 2018. [Online]. Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.

22. "What is Logistic Regression?," Complete dissertation, [Online]. Available: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/.

23. N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Computer Science,* vol. 89, pp. 213-217, 2016.

24. Q. T. Truong, G. Touya and . C. D. Runz, "OSMWatchman: Learning How to Detect Vandalized Contributions in OSM Using a Random Forest Classifier," *Mdpi open access journal,* vol. 9, no. 9, p. 504, 2020.

25. Q. T. Tabassum, G. Ghosh, A. Atika and A. Chakrabarty, "Detecting Online Recruitment Fraud Using Machine Learning," *IEEE,* pp. 472-477, 2021.

26. W. S. Nobel, "What is a support vector machine?," *Nature biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

27. "Confusion matrix for your multi-class machine learning model," Towards data science, 2020. [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

28. "Simple guide to confusion matrix terminology," Data school, 2014. [Online]. Available: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/.

29. A. Uddin, B. Talukder, M. M. Khan and A. Zaguia, "Study on Convolutional Neural Network to Detect COVID-19 from Chest X-Rays," *Hindawi,* 2021.

APPENDIX SOFTWARE LISTING

About Project

First of all, we will implement the required libraries.

```
#Importing Libraries

import re

import string

import numpy as np

import pandas as pd

import random

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.model_selection import train_test_split

from sklearn.pipeline import Pipeline

from sklearn.base import TransformerMixin

from sklearn.metrics import accuracy_score, plot_confusion_matrix, classification_report,
confusion_matrix

from wordcloud import WordCloud

import spacy

from spacy.lang.en.stop_words import STOP_WORDS

from spacy.lang.en import English

from sklearn.svm import SVC
```

After importing the required libraries, we will read the data set to the program.

```
#Reading dataset

data = pd.read_csv('fake_job_postings.csv')
```

#We will check the shape of the dataset and the top five elements of the dataset.

#Shape of the dataset

Data.shape

```
(17880, 18)
```

data.head()

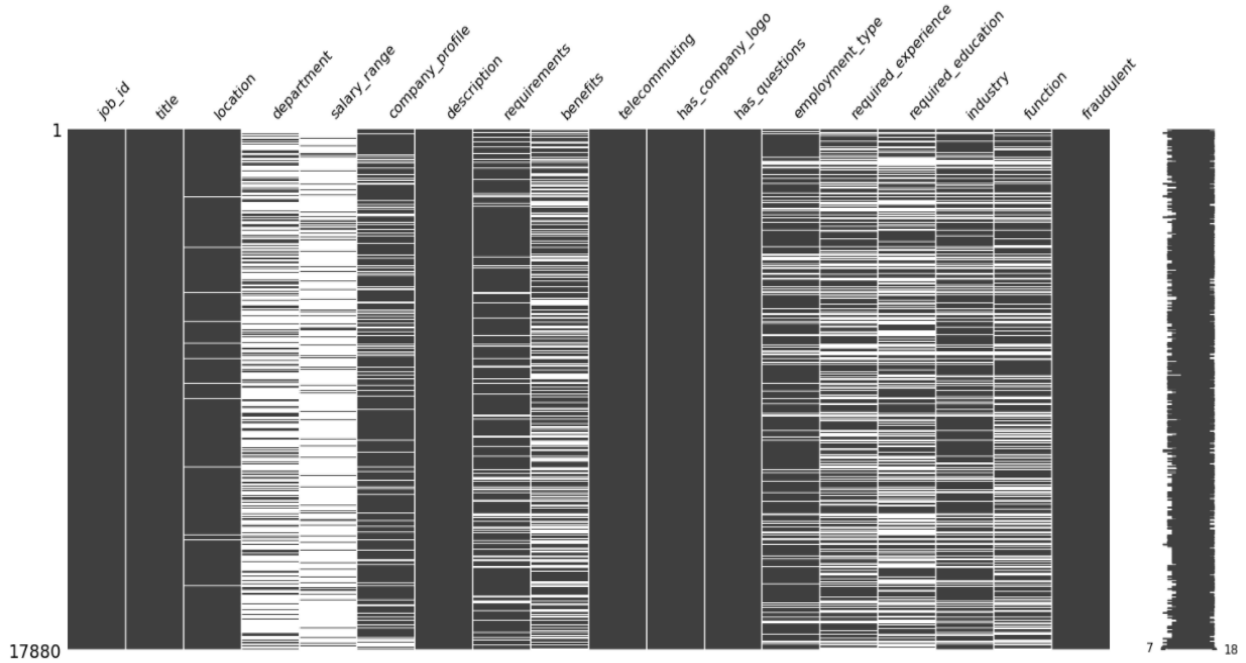
Out[3]:

	job_id	title	location	department	salary_range	company_profile	description	requirements	benefits	telecommuting	has_c
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...	NaN	0	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	0	
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...	NaN	0	
3	4	Account Executive - Washington DC	US, DC, Washington	Sales	NaN	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our culture is anything but corporate—we have ...	0	
4	5	Bill Review Manager	US, FL, Fort Worth	NaN	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Benefits Offered	0	

checking missing data in our dataframe.

missingno.matrix(data)

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa53e70f1d0>




```
print(data.columns)

data.describe()
```

```
Index(['job_id', 'title', 'location', 'department', 'salary_range',
       'company_profile', 'description', 'requirements', 'benefits',
       'telecommuting', 'has_company_logo', 'has_questions', 'employment_type',
       'required_experience', 'required_education', 'industry', 'function',
       'fraudulent'],
      dtype='object')
```

Out[5]:

	job_id	telecommuting	has_company_logo	has_questions	fraudulent
count	17880.000000	17880.000000	17880.000000	17880.000000	17880.000000
mean	8940.500000	0.042897	0.795302	0.491723	0.048434
std	5161.655742	0.202631	0.403492	0.499945	0.214688
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	4470.750000	0.000000	1.000000	0.000000	0.000000
50%	8940.500000	0.000000	1.000000	0.000000	0.000000
75%	13410.250000	0.000000	1.000000	1.000000	0.000000
max	17880.000000	1.000000	1.000000	1.000000	1.000000

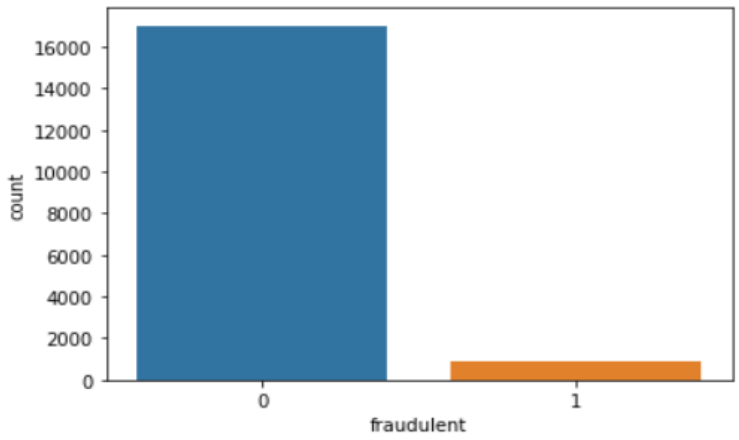
Now lets see how many jobs posted are fraud and real.

sns.countplot(data.fraudulent)

data.groupby('fraudulent').count()['title'].reset_index().sort_values(by='title',ascending=False)

Out[6]:

	fraudulent	title
0	0	17014
1	1	866



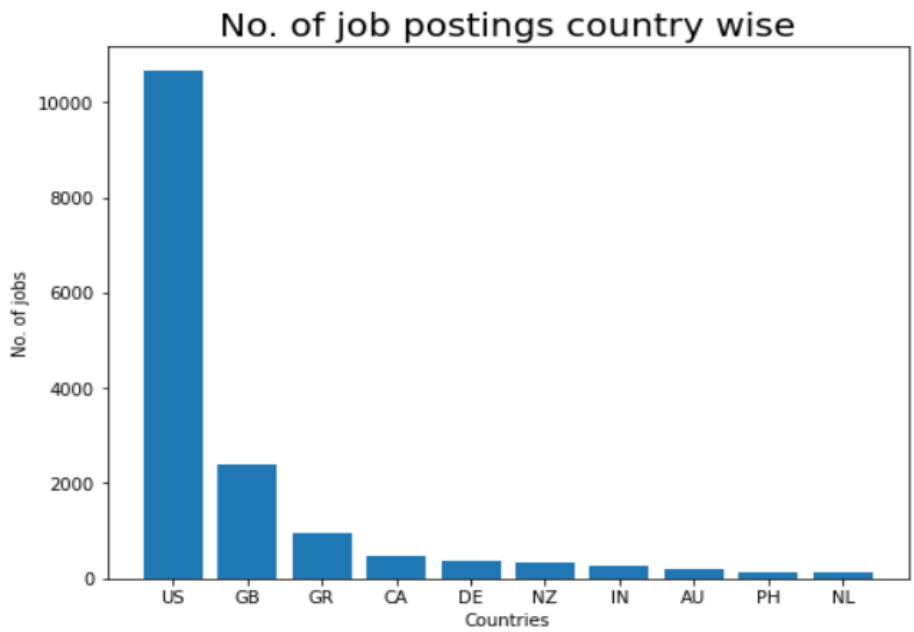

```
columns=['job_id', 'telecommuting', 'has_company_logo', 'has_questions',
'salary_range', 'employment_type']

for col in columns:

    del data[col]


data.fillna(' ', inplace=True)
```

```
data.head()
```

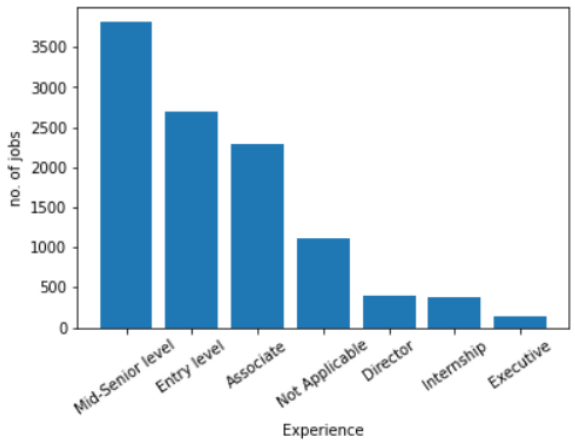
Out[8]:

	title	location	department	company_profile	description	requirements	benefits	required_experience	required_education
0	Marketing Intern	US, NY, New York	Marketing	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...		Internship	
1	Customer Service - Cloud Video Production	NZ, , Auckland	Success	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - AwesomeIDo you...	What we expect from you:Your key responsibilit...	What you will get from usThrough being part of...	Not Applicable	
2	Commissioning Machinery Assistant (CMA)	US, IA, Wever		Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...			
3	Account Executive - Washington DC	US, DC, Washington	Sales	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our culture is anything but corporate—we have ...	Mid-Senior level	Bachelor's Degree
4	Bill Review Manager	US, FL, Fort Worth		SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Benefits Offered	Mid-Senior level	Bachelor's Degree

```
def split(location):
    l = location.split(',')
    return l[0]
data['country'] = data.location.apply(split)

country = dict(data.country.value_counts()[:11])
del country[' ']
plt.figure(figsize=(8,6))
plt.title('No. of job postings country wise', size=20)
plt.bar(country.keys(), country.values())
plt.ylabel('No. of jobs', size=10)
plt.xlabel('Countries', size=10)

Text(0.5, 0, 'Countries')
```




```
experience = dict(data.required_experience.value_counts())

del experience[' ']

plt.bar(experience.keys(), experience.values())

plt.xlabel('Experience', size=10)

plt.ylabel('no. of jobs', size=10)

plt.xticks(rotation=35)

plt.show()
```



```
# title of jobs which are frequent.
print(data.title.value_counts()[:10])
```

```
English Teacher Abroad                                311
Customer Service Associate                            146
Graduates: English Teacher Abroad (Conversational)    144
English Teacher Abroad                                 95
Software Engineer                                      86
English Teacher Abroad (Conversational)                83
Customer Service Associate - Part Time                 76
Account Manager                                        75
Web Developer                                          66
Project Manager                                        62
Name: title, dtype: int64
```

```
data['text']=data['title']+' '+data['location']+' '+data['company_profile']+'
'+data['description']+' '+data['requirements']+' '+data['benefits']
del data['title']
del data['location']
del data['department']
del data['company_profile']
del data['description']
del data['requirements']
del data['benefits']
del data['required_experience']
del data['required_education']
del data['industry']
del data['function']
del data['country']
```

```
data.head()
```

Out[14]:

	fraudulent	text
0	0	Marketing Intern US, NY, New York We're Food52...
1	0	Customer Service - Cloud Video Production NZ, ...
2	0	Commissioning Machinery Assistant (CMA) US, IA...
3	0	Account Executive - Washington DC US, DC, Wash...
4	0	Bill Review Manager US, FL, Fort Worth SpotSou...

```
fraudjobs_text = data[data.fraudulent==1].text

actualjobs_text = data[data.fraudulent==0].text


STOPWORDS = spacy.lang.en.stop_words.STOP_WORDS

plt.figure(figsize = (16,14))

wc = WordCloud(min_font_size = 3,  max_words = 3000 , width = 1600 , height = 800
, stopwords = STOPWORDS).generate(str(" ".join(fraudjobs_text)))

plt.imshow(wc,interpolation = 'bilinear')
```

Out[16]: <matplotlib.figure.AxesImage at 0x7fa534729410>


```
plt.figure(figsize = (16,14))
```

```
wc = WordCloud(min_font_size = 3,  max_words = 3000 , width = 1600 , height = 800
, stopwords = STOPWORDS).generate(str(" ".join(actualjobs_text)))

plt.imshow(wc,interpolation = 'bilinear')


# Create our list of punctuation marks

punctuations = string.punctuation


# Create our list of stopwords

nlp = spacy.load('en')

stop_words = spacy.lang.en.stop_words.STOP_WORDS


# Load English tokenizer, tagger, parser, NER and word vectors

parser = English()


# Creating our tokenizer function

def spacy_tokenizer(sentence):

    # Creating our token object, which is used to create documents with
linguistic annotations.

    mytokens = parser(sentence)



    # Lemmatizing each token and converting each token into lowercase

    mytokens = [ word.lemma_.lower().strip() if word.lemma_ != "-PRON-" else
word.lower_ for word in mytokens ]



    # Removing stop words

    mytokens = [ word for word in mytokens if word not in stop_words and word not
in punctuations ]



    # return preprocessed list of tokens

    return mytokens
```

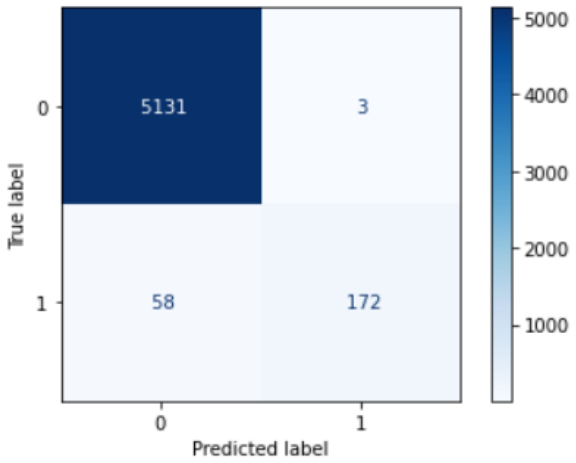
```
# Custom transformer using spaCy
class predictors(TransformerMixin):
    def transform(self, X, **transform_params):
        # Cleaning Text
        return [clean_text(text) for text in X]

    def fit(self, X, y=None, **fit_params):
        return self

    def get_params(self, deep=True):
        return {}

# Basic function to clean the text
def clean_text(text):
    # Removing spaces and converting text into lowercase
    return text.strip().lower()

# creating our bag of words
bow_vector = CountVectorizer(tokenizer = spacy_tokenizer, ngram_range=(1,3))

# splitting our data in train and test
X_train, X_test, y_train, y_test = train_test_split(data.text, data.fraudulent,
test_size=0.3)
```

Logistic Regression

```
clf = LogisticRegression()

# Create pipeline using Bag of Words
pipe1 = Pipeline([("cleaner", predictors()),
```

```
                 ('vectorizer', bow_vector),
                 ('classifier', clf)])

# fitting our model.
pipe1.fit(X_train,y_train)

Pipeline(steps=[('cleaner', <__main__.predictors object at 0x7fa506791910>),
                ('vectorizer',
                 CountVectorizer(ngram_range=(1, 3),
                                 tokenizer=<function spacy_tokenizer at
0x7fa53480ef80>)),
                ('classifier', LogisticRegression())])

# Predicting with a test dataset
predicted = pipe.predict(X_test)

# Model Accuracy
print("Logistic Regression Accuracy:", accuracy_score(y_test, predicted))
print("Logistic Regression Recall:", recall_score(y_test, predicted))

Logistic Regression Accuracy: 0.988627889634601
Logistic Regression Recall: 0.7478260869565218

plot_confusion_matrix(pipe, X_test, y_test, cmap='Blues', values_format=' ')
```

Out[24]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fa5314a09d0>




```
predictions = pipe.predict(X_test)

from sklearn.metrics import classification_report

print("Classification report of Logistic Regression" )

print(classification_report(y_test, predictions))
```

In [43]:
```
predictions = pipe.predict(X_test)
from sklearn.metrics import classification_report
print("Classification report of Logistic Regression" )
print(classification_report(y_test, predictions))
```

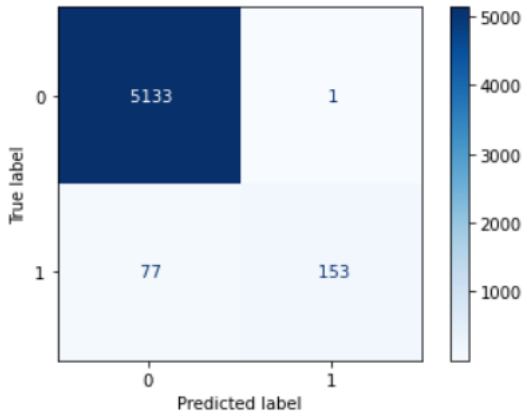
```
Classification report of Logistic Regression
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      5134
           1       0.98      0.75      0.85       230

    accuracy                           0.99      5364
   macro avg       0.99      0.87      0.92      5364
weighted avg       0.99      0.99      0.99      5364
```

Random Forest Classifier

clf = RandomForestClassifier()

```
# Create pipeline using Bag of Words
pipe = Pipeline([("cleaner", predictors()),
                 ('vectorizer', bow_vector),
                 ('classifier', clf)])

# fitting our model.
pipe.fit(X_train,y_train)

Pipeline(steps=[('cleaner', <__main__.predictors object at 0x7fa50c833b50>),
                ('vectorizer',
                 CountVectorizer(ngram_range=(1, 3),
                                 tokenizer=<function spacy_tokenizer at
0x7fa53480ef80>)),
                ('classifier', RandomForestClassifier())])

# Predicting with a test dataset
predicted = pipe.predict(X_test)

# Model Accuracy
print("Random Forest Accuracy:", accuracy_score(y_test, predicted))
print("Random Forest Recall:", recall_score(y_test, predicted))

Random Forest Accuracy: 0.9854586129753915
Random Forest Recall: 0.6652173913043479
```

```
In [27]: plot_confusion_matrix(pipe, X_test, y_test, cmap='Blues', values_format=' ')
Out[27]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fa5330a4e10>
```




```
predictions = pipe.predict(X_test)

from sklearn.metrics import classification_report

print("Classification report of Random Forest Classifier" )

print(classification_report(y_test, predictions))
```

```
Classification report of Random Forest Classifier
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      5134
           1       0.99      0.66      0.79       230

    accuracy                           0.99      5364
   macro avg       0.99      0.83      0.89      5364
weighted avg       0.99      0.99      0.98      5364
```