**CHAPTER 1**

# INTRODUCTION

## BRIEF INTRODUCTION



Fig 1.1: Fake Job Prediction

Employment scams are on the rise. According to CNBC, the number of employment scams doubled in 2018 as compared to 2017. The current market situation has led to high unemployment. Economic stress and the impact of the coronavirus have significantly reduced job availability and the loss of jobs for many individuals.

A case like this presents an appropriate opportunity for scammers. Many people are falling prey to these scammers using the desperation that is caused by an unprecedented incident. Most scammers do this to get personal information from the person they are scamming. Personal information can contain address, bank account details, social security number etc.

I am a university student, and I have received several such scam emails. The scammers provide users with a very lucrative job opportunity and later ask for money in return. Sometimes they require investment from the job seeker with the promise of a job. This is a dangerous problem that can be addressed through Machine Learning techniques and Natural Language Processing (NLP).

This project uses data provided from Kaggle. This data contains features that define a job posting. These job postings are categorized as either real or fake. Fake job postings are a very small fraction of this dataset. That is as excepted. We do not expect a lot of fake jobs postings. This project follows five stages. The five stages adopted for this project are :-

1. Problem Definition (Project Overview, Project statement and Metrics)
2. Data Collection
3. Data cleaning, exploring and pre-processing
4. Modeling
5. Evaluating

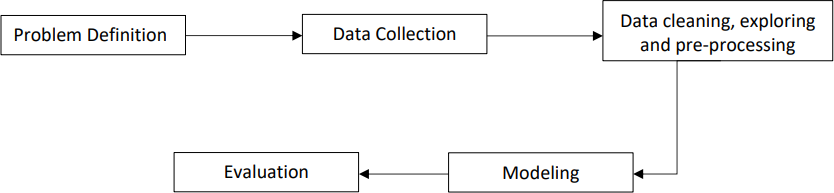


Fig 1.2: Stages of Development

## MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

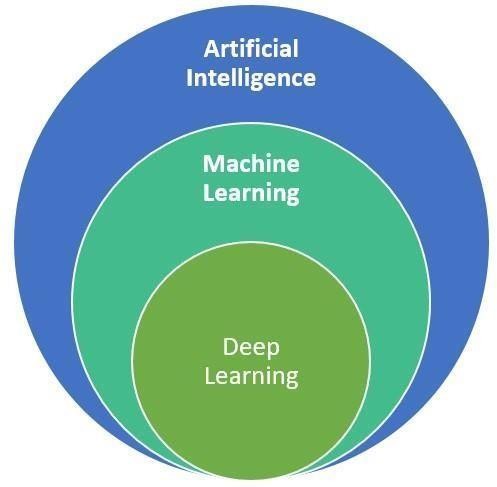


Fig 1.3: Fields of Artificial Intelligence

## MACHINE LEARNING ALGORITHMS

* + - 1. **SUPERVISED LEARNING**

A support-vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white. Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data and consists of a set of training examples.

In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Types of supervised learning algorithms include active learning, classification and regression.

Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data.

## UN-SUPERVISED LEARNING

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labeled. The algorithms are thus allowed to classify, label and/or group the data points contained within the data sets without having any external guidance in performing that task. In other words, unsupervised learning allows the system to identify patterns within data sets on its own.

In unsupervised learning, an AI system will group unsorted information according to similarities and differences even though there are no categories provided. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. Additionally, subjecting a system to unsupervised learning is one way of testing AI.

However, unsupervised learning can be more unpredictable than a supervised learning model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order. AI systems capable of unsupervised learning are often associated with generative learning models, although they may also use a retrieval-based approach. Chatbots, self-driving cars, facial recognition programs, expert systems and robots are among the systems that may use either supervised or unsupervised learning approaches, or both.

In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

## DECISION TREE

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision tree learning uses a decision tree as a predictive model to go from observations about an item to conclusions about the item's target value. It is one of the predictive modeling approaches used in statistics, data mining, and machine learning.

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

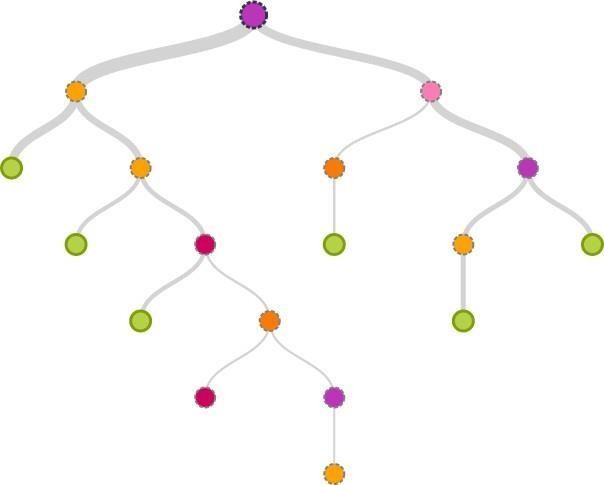


Fig 1.4: Decision Tree

Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees were the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

## STOCASTIC GRADIENT DESCENT

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) [Support Vector](https://en.wikipedia.org/wiki/Support_vector_machine) [Machines](https://en.wikipedia.org/wiki/Support_vector_machine) and [Logistic Regression.](https://en.wikipedia.org/wiki/Logistic_regression) Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features.

Strictly speaking, SGD is merely an optimization technique and does not correspond to a specific family of machine learning models. It is only a *way* to train a model. Often, an instance of [SGD Classifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier) or [SGD Regressor](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html#sklearn.linear_model.SGDRegressor) will have an equivalent estimator in the scikit- learn API, potentially using a different optimization technique. For example, using SGD Classifier results in logistic regression, i.e. a model equivalent to [Logistic Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression) which is fitted via SGD instead of being fitted by one of the other solvers in [Logistic Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression)..

## NAÏVE BAYES

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem. Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

## RANDOM FOREST

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

We need to approach the Random Forest regression technique like any other machine learning technique:

* Design a specific question or data and get the source to determine the required data.
* Make sure the data is in an accessible format else convert it to the required format.
* Specify all noticeable anomalies and missing data points that may be required to achieve the required data.
* Create a machine learning model
* Set the baseline model that you want to achieve
* Train the data machine learning model.
* Provide an insight into the model with test data
* Compare the performance metrics of both the test data and the predicted data from the model.
* If the expectations are not satisfied, you can try improving your model accordingly or dating your data or use another data modeling technique.
* At this stage you interpret the data you have gained and report accordingly.

## BAYESIAN NETWORK

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

## ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another.

An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs.

Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

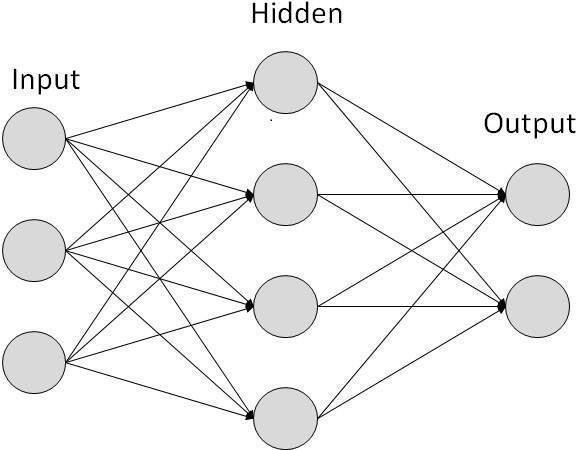


Fig 1.5: A Simple Neural Network

## NATURAL LANGUAGE PROCESSING (NLP)

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

## PROBLEM STATEMENT

This project aims to create a classifier that will have the capability to identify fake and real jobs. The final result will be evaluated based on two different models. Since the data provided has both numeric and text features one model will be used on the text data and the other on numeric data. The final output will be a combination of the two.

The final model will take in any relevant job posting data and produce a final result determining whether the job is real or not.

## OBJECTIVES

The objective of our project is to design a qualified training program so that the competence of job seekers can be in line with the industrial needs, Training Need Analysis (TNA) should be conducted before establish a training program.

But unfortunately there have been many fraudulent job ads by individuals on behalf of the company. In order to overcome the problem, employment scam detection is required. Different classifiers are used for checking fraudulent post in the web and the results of those classifiers are compared for identifying the best employment scam detection model.

## ORGANIZATION OF THE PROJECT REPORT

* + - **Chapter 1:** Discuss the introduction about Machine Learning and various ML Models.
    - **Chapter 2:** Contains the information about the works on the survey.
    - **Chapter 3:** Discuss about the system requirements and describe the functional and non-functional requirements also software and hardware used.
    - **Chapter 4:** Discuss the information about the system architecture and it’s design.
    - **Chapter 5:** Discuss briefly about the functionalities and the various implemented models.
    - **Chapter 6:** Discuss the different models and its results. Also, the snapshots of the output are included.
    - **Chapter 7:** Discuss conclusions made from the results and include future improvements.

**CHAPTER 2**

**LITERATURE** **SURVEY**

## LITERATURE REVIEWS

To get the required knowledge about various concepts related to the present application, existing literature surveys were studied. Some of the important conclusions were made through those are listed below.

### “Detecting Online Recruitment Fraud Using Machine Learning” Hridita Tabassum; Gitanjali Ghosh; Afra Atika; Amitabha Chakrabarty, Member, IEEE [2021]

Online Recruitment fraud (ORF) is becoming an important issue in the cyber-crime region. Companies find it easier to hire people with the help of the internet rather than the old traditional way. But it has greatly attracted scammers.

In this paper, we have proposed a solution on how to detect ORF. We have presented our results based on the previous model and the methodologies, to create the ORF detection model where we have used our own dataset. We have created our dataset based on the Bangladesh job field and by using a publicly accessible dataset as a reference.

### “Online Recruitment Fraud Detection using ANN”Ibrahim M. Nasser; Amjad

**H. Alzaanin; Ashraf Yunis Maghari, Member, IEEE [2021]**

Online recruitment provides job-seekers an efficient search and reach for jobs. It also helps recruiters searching for qualified candidates which improves the recruitment process. However, employment scam has emerged as a critical issue. Some job posts are legitimate, and others are fraud.

In this paper, an Artificial Neural Network based model is proposed to detect fraud job posts. The public Employment Scam Aegean Dataset (EMSCAD) is used with proper text preprocessing techniques for training and testing the proposed model. Our model has precision, recall, and f-measure of 91.84%, 96.02%, and 93.88% respectively. The results show that the proposed ANN-based model outperforms similar existing models in detecting fraud jobs.

### “Fake Job Recruitment Detection Using Machine Learning Approach” Shawni Dutta and Prof. Samir Kumar Bandyopadhyay, Lecturer, The Bhowanipur Education Society College, Member, IEEE [2020]

To avoid fraudulent post for job in the internet, an automated tool using machine learning based classification techniques is proposed in the paper. Different classifiers are used for checking fraudulent post in the web and the results of those classifiers are compared for identifying the best employment scam detection model. It helps in detecting fake job posts from an enormous number of posts. Two major types of classifiers, such as single classifier and ensemble classifiers are considered for fraudulent job posts detection. However, experimental results indicate that ensemble classifiers are the best classification to detect scams over the single classifiers.

Employment scam detection will guide job-seekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for employment scam detection.

### “Fake Job Detection and Analysis Using Machine Learning and Deep Learning Algorithms” C.S. Anita ; P. Nagarajan; G. Aditya Sairam; P. Ganesh; G. Deepak Kumar, Professor, R.M.D. Engineering College, Kavaraipettai, Tamil Nadu, India, Member, IEEE [2020]

In this paper, we have applied machine learning and deep learning algorithms to classify and detect fake jobs from real jobs in a large dataset of job posts. Machine learning algorithms such as logistic regression, KNN classifier and random forest algorithm are used for classification purpose. Deep learning algorithm, Bi-Directional LSTM is used to train the neurons for classification.

### "An Intelligent Model for Online Recruitment Fraud Detection"-[Journal of Information Security, 2019, 10, 155-176]

This study research attempts to prohibit privacy and loss of money for individuals and organization by creating a reliable model which can detect the fraud exposure in the online recruitment environments. This research presents a major contribution represented in a reliable detection model using ensemble approach based on Random forest classifier to detect Online Recruitment Fraud (ORF).

The detection of Online Recruitment Fraud is characterized by other types of electronic fraud detection by its modern and the scarcity of studies on this concept. The researcher proposed the detection model to achieve the objectives of this study. For feature selection, support vector machine method is used and for classification and detection, ensemble classifier using Random Forest is employed.

A freely available dataset called Employment Scam Aegean Dataset (EMSCAD) is used to apply the model. Pre-processing step had been applied before the selection and classification adoptions. The results showed an obtained accuracy of 97.41%. Further, the findings presented the main features and important factors in detection purpose include having a company profile feature, having a company logo feature and an industry feature.

### "Fraud Detection using Machine Learning and Deep Learning" Pradheepan Raghavan; Neamat El Gayar, Member, IEEE [2020]

Frauds are known to be dynamic and have no patterns, hence they are not easy to identify. Fraudsters use recent technological advancements to their advantage. They somehow bypass security checks, leading to the loss of millions of dollars. Analyzing and detecting unusual activities using data mining techniques is one way of tracing fraudulent transactions. This paper aims to benchmark multiple machine learning methods such as k-nearest neighbor (KNN), random forest and support vector machines (SVM), while the deep learning methods such as auto encoders, convolutional neural networks (CNN), restricted boltzmann machine (RBM) and deep belief networks (DBN). The datasets which will be used are the European (EU) Australian and German dataset. The Area Under the ROC Curve (AUC), Matthews Correlation Coefficient (MCC) and Cost of failure are the 3-evaluation metrics that would be used.

### "Fraudlent Jobs Detection in Recruitment Domain using Knowledge Graphs”, IEEE [2021]

Fraudulent jobs are an emerging threat over online recruitment platforms such as LinkedIn, Glassdoor. Fraudulent job postings affect the platform’s trustworthiness and have a negative impact on user experience. Therefore, these platforms need to detect and remove these fraudulent jobs. Generally, fraudulent job postings contain untenable facts about domain- specific entities such as mismatch in skills, industries, offered compensation, etc. However, existing approaches focus on studying writing styles, linguistics, and context-based features, and ignore the relationships among domain-specific entities. To bridge this gap, we propose an approach based on the Knowledge Graph (KG) of domain-specific entities to detect fraudulent jobs.

In this paper, we present a multi-tier novel end-to-end framework called FRaudulent Jobs Detection (FRJD) Engine, which considers a) fact validation module using KGs, b) contextual module using deep neural networks c) meta-data module to capture the semantics of job postings. We conduct our experiments using a fact validation dataset containing 4 million facts extracted from job postings. Extensive evaluation shows that FRJD yields a 0.96 F1-score on the curated dataset of 157,880 job postings. Finally, we provide insights on the performance of different fact-checking algorithms on recruitment domain datasets.

## PROBLEM SATEMENT

This project aims to create a classifier that will have the capability to identify fake and real jobs. The final result will be evaluated based on two different models. Since the data provided has both numeric and text features one model will be used on the text data and the other on numeric data. The final output will be a combination of the two.

The final model will take in any relevant job posting data and produce a final result determining whether the job is real or not.

## EXISTING SYSTEM

* + 1. **Description**

The proposed system based on K-nearest Neighbor classifier which gives accuracy less than 85% and Decision Tree Classifier which gives the accuracy less than 95% that does not require heavy pre-processing. The proposed method performed robustly.

## Drawback

In this, we could see that the software made using the above mentioned classifiers gave the accuracy of less than 85%and 95% respectively. This may be something we need to solve in future work by using classifiers like ANN, Logistic Regression and Random Forest classifier to increase the accuracy more than 95%.

## PROPOSED SYSTEM

* + 1. **Description**
* This project aims to create classifier that will have the capability to identify fake and real jobs.
* The final result will be evaluated based on two different models. Since the data provided has both numeric and text features one model will be used on the text data and other on numeric data.
* The final output will be a combination of the two. The final model will take in any relevant job posting and produce a final result determining whether the job is real or not.

## Advantages

* Accuracy of different prediction models, where Light GBM (95.17%) and Gradient Boosting (95.17%) give the highest accuracy.
* ANN model has precision, recall, and f-measure of 91.84%, 96.02%, and 93.88% respectively.
* Accuracy of different prediction models, where Logistic Regression has 93% accuracy and random forest has 97% accuracy.

**CHAPTER 3**

# SYSTEM REQUIREMENTS

A System Requirement Specification (SRS) is basically an organization’s understanding of a customer or potential client’s system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives a brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It’s a two-way insurance policy that assures that both the client and the organization understand the other’s requirements from that perspective at a given point in time.

## FUNCTIONAL REQUIREMENTS

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing, and other specific functionality. In this system following are the functional requirements; =

* The control is provided to the respective authority.
* When a fake job is predicted by the software, it alarms the user.
* The authority takes the necessary actions which is required when the alarm is raised.

## NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the application. They may relate to emergent properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the application interface. Many non- functional requirements related to the system as whole rather than to individual application feature. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention.

* + **Usability:** The application should have a user-friendly interface that will require minimal training before the user starts using it.
  + **Performance:** The application should work at an optimum speed with the response time of the entire application should be at acceptable level.
  + **Reliability:** The application should provide services as specified in the functional requirements each time it's been run.
  + **Scalability:** The capability of a system, network or process to handle a growing amount of work, or its potential to be enlarged in order to accommodate the growth.

## SOFTWARE/ HARDWARE USED

### HARDWARE REQUIREMENTS

In this system following are the software requirements:

* + - * System : Laptop/ Desktop
      * Speed : 2.3 GHz
      * RAM : 4 GB
      * Hard Disk : 32 GB
      * Modem (Wi-Fi / Ethernet)
      * I/O Devices

### SOFTWARE REQUIREMENTS

The software requirements are the description of the features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected, or unexpected from the client’s point of view. In this system following are the software requirements:

* + - * Operating System : Windows 7 or above
      * Dataset : Imported from Kaggle
      * Deep Learning Model : GCD, Naïve-Bayes
      * Python : 3.6+
      * Packages/Modules : Pandas, Seaborn, Numpy, Matplotlib, sklearn, skimage.
* In this project, we proposed a solution on how to detect ORF. They presented results based on the previous model and the methodologies, to create the ORF detection model where they used own dataset.
* K-nearest Neighbour Classifier, Logistic Regression, Ada Boost, Decision Tree Classifier, Random Forest Classifier, Voting Classifier, Light GBM, Gradient Boosting are the algorithms that have been used.
* The public Employment Scam Aegean Dataset (EMSCAD) is used with proper text pre- processing techniques for training and testing the proposed model.

**CHAPTER 4**

# SYSTEM DESIGN / METHODOLOGY

## ARCHITECTURE

A system architecture is a conceptual model using which we can define the structure and behavior of that system. It is a formal representation of a system. Depending on the context, the system architecture can be used to refer to either a model to describe the system or a method used to build the system. Building a proper system architecture helps in the analysis of the project, especially in the early stages.

A diagrammatic representation of the implementation of this project is given below. The dataset is split into text, numeric and y-variable. The text dataset is converted into a term-frequency matrix for further analysis. Then using sci-kit learn, the datasets are split into test and train datasets. The baseline model Naïve bayes and another model SGD is trained on the using the train set which is 70% of the dataset. The final outcome of the models based on two test sets – numeric and text are combined such that if both models say that a particular data point is not fraudulent only then a job posting is fraudulent. This is done to reduce the bias of Machine Learning algorithms towards majority classes. The trained model is used on the test set to evaluate model performance. The Accuracy and F1-score of the two models – Naïve bayes and SGD are compared and the final model for our analysis is selected.

Employment scam detection will guide job-seekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for employment scam detection. Experimental results indicate that Random Forest classifier outperforms over its peer classification tool. The proposed approach achieved accuracy 98.27% which is much higher than the existing methods.

## RECOMMENDATION FUNCTIONALITY

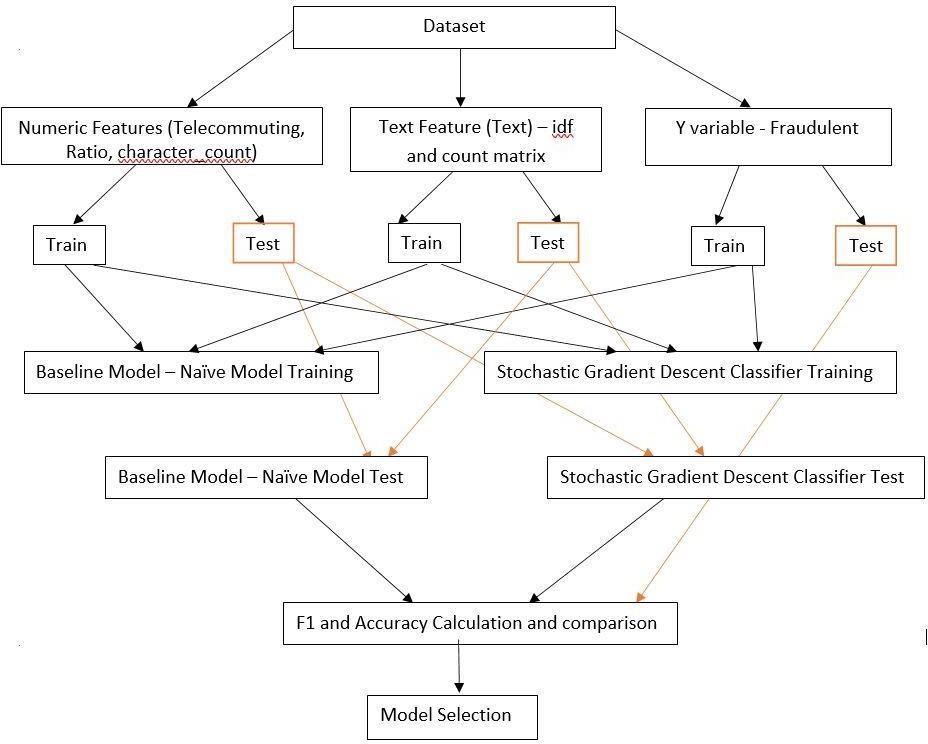


Fig 4.1: System Architecture of proposed system.

## METRICS

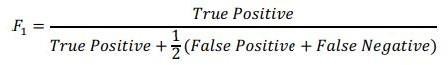
The models will be evaluated based on two metrics:

1. Accuracy: This metric is defined by this formula –



As the formula suggests, this metric produces a ratio of all correctly categorized data points to all data points. This is particularly useful since we are trying to identify both real and fake jobs unlike a scenario where only one category is important. There is however one drawback to this metric. Machine learning algorithms tend to favor dominant classes. Since our classes are highly unbalanced a high accuracy would only be a representative of how well our model is categorizing the negative class (real jobs).

1. F1-Score: F1 score is a measure of a model’s accuracy on a dataset. The formula for this metric is –



F1-score is used because in this scenario both false negatives and false positives are crucial. This model needs to identify both categories with the highest possible score since both have high costs associated to it.

## DATA PREPROCESSING

The following steps are taken for text processing:



Fig 4.5: Text Processing

* Tokenization: The textual data is split into smaller units. In this case the data is split into words.
* To Lower: The split words are converted to lowercase
* Stopword removal: Stopwords are words that do not add much meaning to sentences. For example: the,

a, an, he, have etc. These words are removed.

* Lemmatization: The process of lemmatization groups in which inflected forms of words are used together.

**CHAPTER 5**

# IMPLEMENTATION

## Overview of technologies Used Naïve Bayes:

Naïve Bayes is an applied math classification approach supporting Bayes mathematical Theorem. It is one among the only and the simplest supervised machine learning (ML) algorithms. This classifier is quick, correct, precise and attested formula. This classifier has high success rate and is fast on massive dataset.

NB classifier presumes that the result of a specific attribute in a very division is a freelance of alternative attribute. As a sample, a person in debt fascinates or not counting over his/her financial gain, old debt and money antiquity, age, place of residence. Although of these options are dependent on each other, these options are still seen as independently. This presumption concludes computation, and that's why it is thought about as NB. This presumption is called as conditional class independence assumption.

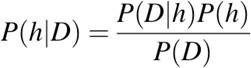


Fig 5.1: Naïve Bayes

* + - P(h): This This is the prior probability of h and is also called as probability of hypothesis of h being true.
    - P(D): This is the prior probability of data.
    - P(h|D): This is posterior probability and is also called as probability of hypothesis h showing the data D
    - P(D|h): This is prior probability and the probability of dataset d given the hypothesis h was True.

## SGD (STOCASTIC GRADIENT DESCENT):

**Stochastic Gradient Descent (SGD)** could be a straightforward and lot economical approach to accommodate regressors underneath protrusive loss parameter like (linear) SVM and Logistical Regression and linear classifier. Even though SGD has been used in Artificial Intelligence and Machine Learning community for such a vast period. It was recognized a major approach in context large -scale learning. SGD classifier has been applied to large-scale and results were shows massive success and distributed ML issues typically encountered in text categorization and NLPs. Given that the dataset is scattered, infrequent, the classifier during this library easily projects to problems with greater than 10,000 training examples and greater than 10,000 options.

The class S-gradient descent Classifier applies a plain stochastic gradient descent learning model which supports different loss parameter and sanction for classification. Down is that the call border of a SGD Classifier trained with the loss, equivalent to a linear SVM.

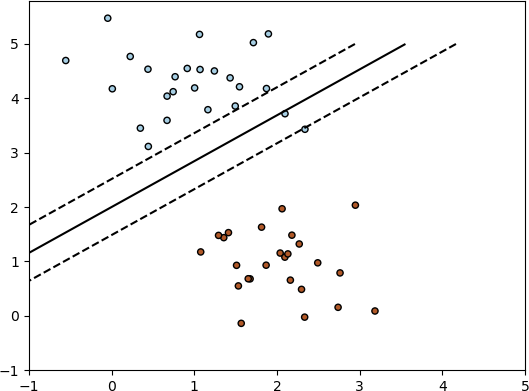


Fig 5.2: SGD Classifier Working

## Data Exploration

The data for this project is available at Kaggle - <https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting> prediction. The dataset consists of 17,880 observations and 18 features.

## The data is combination of integer, binary and textual datatypes. A brief definition of the variables is given below:

Table 1: Table of Variables

## 

## 

Since most of the datatypes are either Booleans or text a summary statistic is not needed here. The only integer is job\_id which is not relevant for this analysis. The dataset is further explored to identify null values.



Fig 5.3: Missing Values

Variables such as department and salary\_range have a lot of missing values. These columns are dropped from further analysis.

After initial assessment of the dataset, it could be seen that since these job postings have been extracted from several countries the postings were in different languages. To simplify the process this project uses data from US based locations that account for nearly 60% of the dataset. This was done to ensure all the data is in English for easy interpretability. Also, the location is split into state and city for further analysis. The final dataset has 10593 observations and 20 features.

The dataset is highly unbalanced with 9868 (93% of the jobs) being real and only 725 or 7% of the jobs being fraudulent.

## Implementation Details of Modules

The dataset is split into text, numeric and y-variable. The text dataset is converted into a term-frequency matrix for further analysis. Then using sci-kit learn, the datasets are split into test and train datasets.

The baseline model Naïve bayes and another model SGD is trained on the using the train set which is 70% of the dataset. The final outcome of the models based on two test sets

– numeric and text are combined such that if both models say that a particular data point is not fraudulent only then a job posting is fraudulent. This is done to reduce the bias of Machine Learning algorithms towards majority classes.

The trained model is used on the test set to evaluate model performance. The Accuracy and F1-score of the two models – Naïve bayes and SGD are compared and the final model for our analysis is selected.

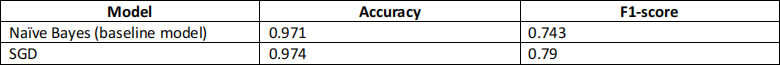
**CHAPTER 6**

# RESULTS AND DISCUSSIONS

## MODEL EVALUATION AND VALIDATION

The final model used for this analysis is – SGD. This is based on the results of the metrics as compared to the baseline model. The outcome of the baseline model and SGD are presented in the table below:

Table 2: Model Comparison



Based on these metrics, SGD has a slightly better performance than the baseline model, Naïve Bayes. This is how the final model is chosen to be SGD and is better.

## JUSTIFICATION

As mentioned above, the final model performs better than the established benchmark of the baseline model. The model will be able to identify real jobs with a very high accuracy. However, it’s identification of fake jobs can still be improved upon.

## RESULTS AND SNAPSHOTS

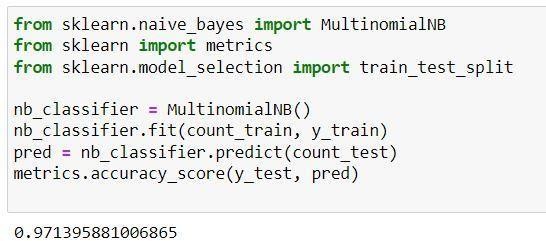


Fig 6.1: Accuracy of Naïve Bayes Model

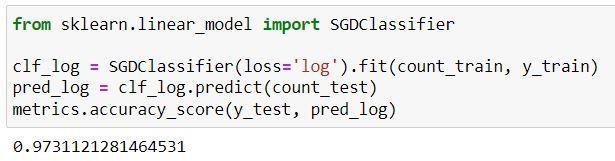


Fig 6.2: Accuracy of SGD Model

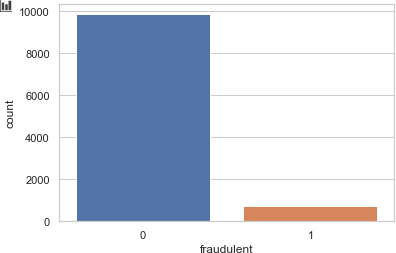


Fig 6.3: Count Plot Real and Fake Jobs

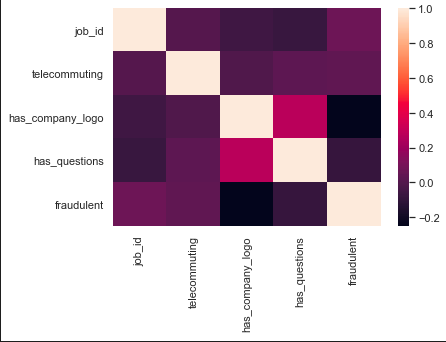


Fig 6.4: Correlation Matrix (Data Visualization)

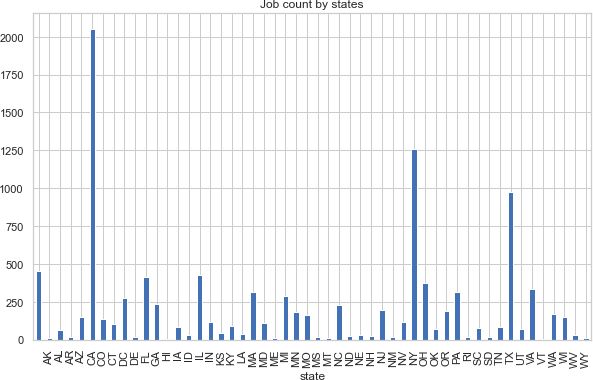


Fig 6.5: Job Count by States

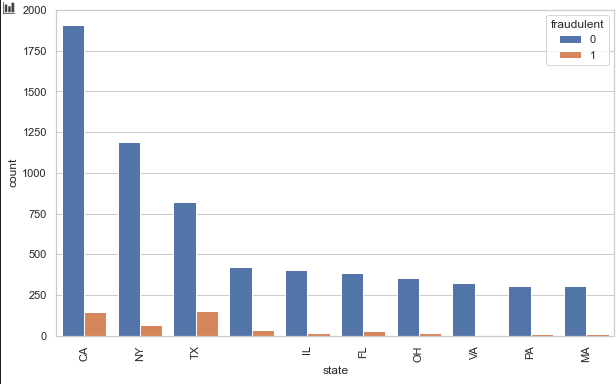


Fig 6.6: Distribution of fake and Real Jobs based on Location

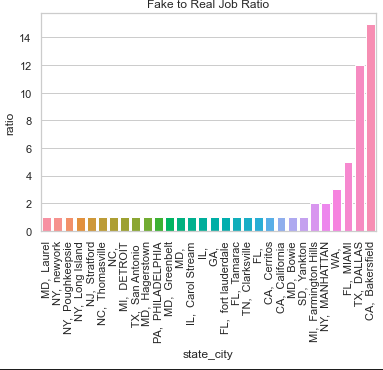
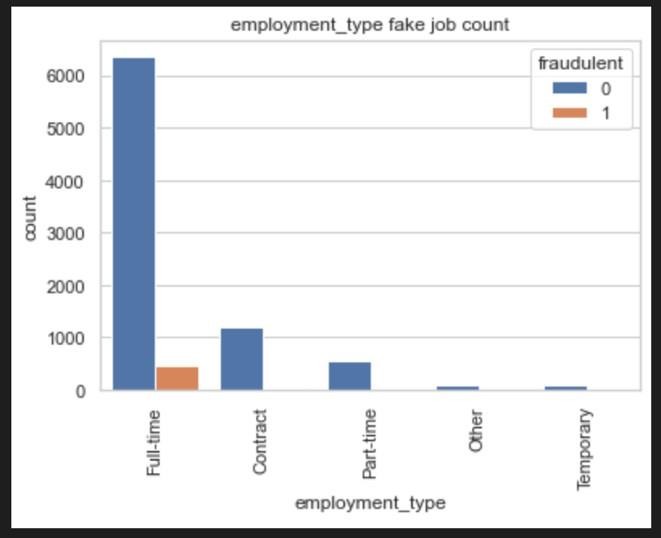
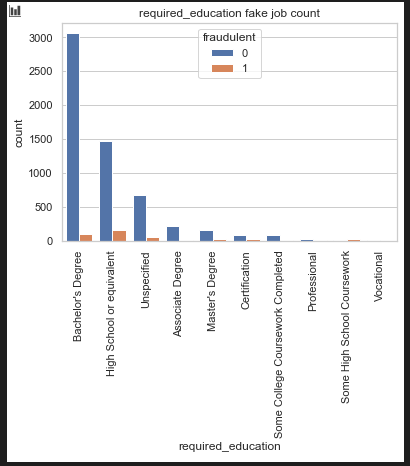


Fig 6.7: Ratio of Fake to Real based on city and state

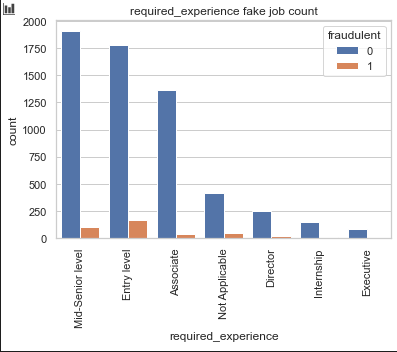


Fig 6.8: Job count based on (a)employment type, (b)Required education, (c)Experience

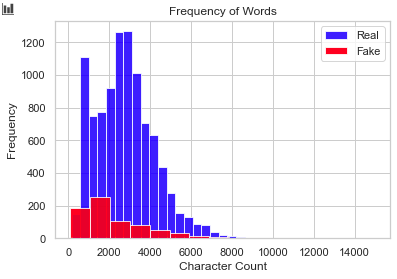


Fig 6.9: Character Count

**CHAPTER 7**

# CONCLUSION

A confusion matrix can be used to evaluate the quality of the project. The project aims to identify real and fake jobs.

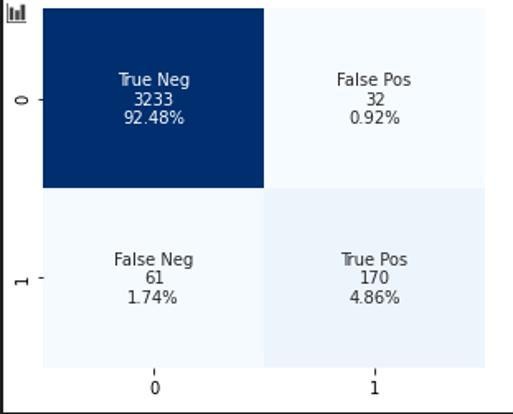


Fig 7.1: Confusion Matrix for the final model

The confusion matrix above displays the following values – categorized label, number of data points categorized under the label and percentage of data represented in each category. The test set has a total of 3265 real jobs and 231 fake jobs. Based on the confusion matrix it is evident that the model identifies real jobs 99.01% of the times. However, fraudulent jobs are identified only 73.5% of the times. Only 2% of the times has the model not identified the class correctly. This shortcoming has been discussed earlier as well as Machine Learning algorithms tend to prefer the dominant classes.

The dataset that is used in this project is very unbalanced. Most jobs are real, and few are fraudulent. Due to this, real jobs are being identified quite well. Certain techniques like SMOTE can be used to generate synthetic minority class samples. A balanced dataset should be able to generate better results.

**FUTURE ENHANCEMENT**

Future work includes implementing the calculation of other data apart from text and numeric data so that the program can analyze natural language to provide more effective and efficient output. It may also include development of a user-friendly mobile application to provide an easy accessibility option for the users.

***REFERENCES***

## Review through Research Papers

1. “Detecting Online Recruitment Fraud Using Machine Learning” Hridita Tabassum; Gitanjali Ghosh; Afra Atika; Amitabha Chakrabarty, Member, IEEE [2021]
2. “Online Recruitment Fraud Detection using ANN”Ibrahim M. Nasser; Amjad H. Alzaanin; Ashraf Yunis Maghari, Member, IEEE [2021]
3. “Fake Job Recruitment Detection Using Machine Learning Approach” Shawni Dutta and Prof. Samir Kumar Bandyopadhyay, Lecturer, The Bhowanipur Education Society College, Member, IEEE [2020]
4. “Fake Job Detection and Analysis Using Machine Learning and Deep Learning Algorithms” C.S. Anita ; P. Nagarajan; G. Aditya Sairam; P. Ganesh; G. Deepak Kumar, Professor, R.M.D. Engineering College, Kavaraipettai, Tamil Nadu, India, Member, IEEE [2020]
5. "An Intelligent Model for Online Recruitment Fraud Detection"-[Journal of Information Security, 2019, 10, 155-176]
6. "Fraud Detection using Machine Learning and Deep Learning" Pradheepan Raghavan; Neamat El Gayar, Member, IEEE [2020]
7. “Fraudlent Jobs Detection in Recruitment Domain using Knowledge Graphs”, IEEE [2021]
8. "A comparative study on fake job post prediction using different data mining techniques", 2021 2nd International Conference on Robotics Electrical and Signal Processing Techniques (ICREST)
9. "Fake job prediction using sequential network", 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)
10. "Orfdetector: Ensemble learning based online recruitment fraud detection", 2019 Twelfth International Conference on Contemporary Computing (IC3)
11. "An intelligent model for online recruitment fraud detection", Journal of Information Security.
12. "Employee recruitment fraud prevention with the implementation of decision support system", Journal of Physics: Conference Series.
13. "Automatic detection of online recruitment frauds: Characteristics methods and a public dataset", Future Internet.
14. "Smart fraud detection framework for job recruitments", Arabian Journal for Science and Engineering.
15. "The gradient boosting algorithm and random boosting for genome-assisted evaluation in large data sets", Journal of dairy science.
16. “A Review on Evaluation Metrics for Data Classification Evaluations,” Int. J. Data Min. Knowl. Manag. Process.
17. [Real or Fake] Fake JobPosting Prediction,Version 1.Retrieved March 29,2020.
18. “An Intelligent Model for Online Recruitment Fraud Detection,” J. Inf. Secur.
19. “Fake Job Recruitment Detection Using Machine Learning Approach”. International Journal of Engineering Trends and Technology (IJETT)
20. “Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset”. Future Internet

## Review through Websites

* 1. https://towardsdatascience.com/
  2. https://en.wikipedia.org/wiki/Fake\_job\_prediction
  3. https:/[/www.youtube.com/](http://www.youtube.com/)

## STUDENT DETAILS

|  |  |
| --- | --- |
| NAME- PRANAV PARTH  USN- 1SG18CS072  SEM, SEC- 8th , B  PHONE NUMBER- 9380452430  EMAIL ID- [pranav.parth01@gmail.com](mailto:pranav.parth01@gmail.com) | NAME- RISHU RAJ  USN- 1SG18CS085  SEM, SEC- 8th , B  PHONE NUMBER- 7479569569  EMAIL ID- rishustyleofjoy@gmail.com |
| NAME- RITIK SAINI  USN- 1SG18CS086  SEM, SEC- 8th, B  PHONE NUMBER-8294483340  EMAIL ID- ritiksainimarch2000@gmail.com | NAME- SWAPNIL  USN-1SG18CS119  SEM, SEC- 8th, B  PHONE NUMBER-9599545052  EMAIL ID- swapnilkr099@gmail.com |