A Project-II Report on

# FAKE JOB RECRUITMENTS DETECTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirement
For
The award of degree of

**BACHELOR OF TECHNOLOGY**

In
**COMPUTER SCIENCE & ENGINEERING**
**2018-2022**

Submitted By

| | |
|---|---|
| MANIDEEP | 16311A05U5 |
| RAJ HEMANI | 16311A05X0 |
| MANISH KUMAR | 17311A05V6 |

Under the Guidance of

Mr.Meeravali Shaik
Assistant Professor

**Department of Computer Science & Engineering**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(AUTONOMOUS)**
Yamnampet (V), Ghatkesar (M), Hyderabad – 501301, A1147.P.

**AY-2021-2022**

**Department of Computer Science and Engineering**

**Sreenidhi Institute of Science and Technology**

## CERTIFICATE

This is to certify that this Project-II report on "**FAKE JOB RECRUITMENT DETECTION USING MACHINE LEARING**," submitted by **MANIDEEP(16311A05U5), RAJ(16311A05X0), MANISH KUMAR(17311A05V6),** in the year 2022 in the partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide work that has been carried out by them as a part of their **Project-II during Fourth Year Second Semester,** under our guidance. This report has not been submitted to any other Institute or university for the award of any degree.

| **Internal guide** | **Project Coordinator** | **Head of the Department** |
|---|---|---|
| Mr. Meeravali Shaik | Mrs. Shivani | Dr. Aruna Varanasi |
| Assistant Professor | Assistant Professor | Professor & HOD |
| Department of CSE | Department of CSE | Department of CSE |

**Signature of the External Examiner**

Date:

## DECLARATION

We,**MANIDEEP(16311A05U5),RAJHEMANI(16311A05X0),MANISHKUMAR (17311A05V6).**students of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR,** studying IV-year II semester, **COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the Project-II work, titled "**PRINTER SERVICE APPLICATION"** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree

## ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably runs the institution and has had the major hand in enabling me to do my project.

I profoundly thank **Dr. Aruna Varanasi**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank our Coordinator **Mrs. Shivani**& my internal guide **Mr .Meeravali Shaik** for Project-II for their technical guidance, constant guidelines, encouragement and support in carrying out my project on time at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

| | |
|---|---|
| **MANIDEEP** | **16311A05U5** |
| **RAJ HEMANI** | **16311A05X0** |
| **MANISH KUMAR** | **17311A05V6** |

## ABSTRACT

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.The critical process of hiring has relatively recently been ported to the cloud. Specifically, the automated systems responsible for completing the recruitment of new employees in an online fashion, aim to make the hiring process more immediate, accurate and cost-efficient. However, the online exposure of such traditional business procedures has introduced new points of failure that may lead to privacy loss for applicants and harm the reputation of organizations. So far, the most common case of Online Recruitment Frauds (ORF), is employment scam. Unlike relevant online fraud problems, the tackling of ORF has not yet received the proper attention, remaining largely unexplored until now. Responding to this need, the work at hand defines and describes the characteristics of this severe and timely novel cyber security research topic. At the same time, it contributes and evaluates the first to our knowledge publicly available dataset of 17,880 annotated job ads, retrieved from the use of a real-life system.

## 1. INTRODUCTION

Employment scam is one of the serious issues in recent times addressed in the domain of Online Recruitment Frauds (ORF) [1]. In

recent days, many companies prefer to post their vacancies online so that these can be accessed easily and timely by the job-seekers. However, this intention may be one type of scam by the fraud people because they offer employment to job-seekers in terms of taking money from them. Fraudulent job advertisements can be posted against a reputed company for violating their credibility. These fraudulent job post detection draws a good attention for obtaining an automated tool for identifying fake jobs and reporting them to people for avoiding applicationfor such jobs.

For this purpose, machine learning approach is applied which employs several classification algorithms for recognizing fake posts. In this case, a classification tool isolates fake job posts froma larger set of job advertisements and alerts the user. To address the problem of identifying scams on job posting, supervised learning algorithm as classification techniques are considered initially. A classifier maps input variable to target classes by considering training data. Classifiers addressed in the paper for identifying fake job posts from the others are described briefly. These classifiers-based predictions may be broadly categorized into -Single Classifier based Prediction and Ensemble Classifiers based Prediction.

If **one** looks hard enough, they can spot the differences between these fake postings and genuine ones. Most of the time these postings don't have a company logo on these postings, the initial response from the company is from an unofficial email account, or during an interview they might ask you for personal confidential information such as your credit card details by saying they need it for personnel verification. In normal economic conditions, all these are evident hints that there something suspicious about the company, but these are not normal economic conditions. These are the worst times we all have seen in our lifetimes, and at this time, desperate individuals just need a job, and by this, these individuals are directly playing into the hands of these scammers. Following is a quote by Daniel Linskey, a former superintendent-in-chief of the Boston Police Department

### 1.1 Motivation

In today's context the majority of the companies are recruiting from online platforms. Therefore, many of the applicants are suffering from the fake job posts resulting in losing their data and money. The applicants are facing difficulties in identifying the fake job posts. so, in order to identify the fake job posts we had come up with a machine learning model which analyses the description of the job posts and predicts weather the job post is fake or not.

### 1.2 Problem Definition

Employment scam is one of the serious issues in recent times addressed in the domain of Online Recruitment Frauds (ORF) . In recent days, many companies prefer to post their vacancies online so that these can be accessed easily and timely by the job-seekers. However, this intention may be one type of scam by the fraud people because they offer employment to job-seekers in terms of taking money from them. Fraudulent job advertisements can be posted against a reputed company for violating their credibility. These fraudulent job post detection draws a good attention for obtaining an automated tool for identifying fake jobs and reporting them to people for avoiding application for such jobs.

### 1.3 Objective

Most of the time these postings don't have a company logo on these postings, the initial response from the company is from an unofficial email account, or during an interview they might ask you for personal confidential information such as your credit card details by saying they need it for personnel verification. In normal economic conditions, all these are evident hints that there something suspicious about the company, but these are not normal economic conditions. These are the worst times we all have seen in our lifetimes, and at this time, desperate individuals just need a job, and by this, these individuals are directly playing into the hands of these scammers.

To create awareness among the job seekers regarding the seriousness of this issue and how, through a machine learning model, we can predict whether a job posting is fraudulent or not. By exploring the data and providing insights into which industries are more affected and what are the critical red flags which can give away these fake postings. By applying machine learning models to predict how we can detect these counterfeit postings

**Challenges:**

Challenges are a big factor that will have a negative impact on the current endeavour. The following are some of the difficulties that fake job recruitments prediction system may face:

- Choosing appropriate dataset, after choosing dataset tuning of the parameters which makes project more efficient to get the desired results.
- The model must be trained with less computing efficiency and power in mind.

# 4 Signs of a fake job advert

### Poor spelling and grammar

A job ad riddled with errors is definitely a red flag. Serious companies will likely have done some checks before publishing, as a poorly written job ad can hurt their public image - something that wouldn't concern a fraudster...

### Requests for money

Any demand for money is a bad sign. If an ad requests an upfront fee, then I'd walk away. The biggest alarm bell should be the method of payment though. Scammers will often ask for e-money, as it is difficult to trace.

### Personal email address

Would you put your career in the hands of a Hotmail email address? Hotmail, Gmail and Yahoo accounts are more typically for personal use, so be careful if any of these pop up. Most companies tend to have their own domain.

### Can't be found online

Look for any online presence. If you're having a hard time tracking them down, it might be because they don't really exist. Some fraudsters will create websites or social accounts though, so don't see this as a surefire sign.

www.agencycentral.co.uk

## 2. LITERATURE REVIEW

### 2.1 Related Work

In this Project We use publicly available dataset shared by Vidros et al. [5]. We next extract 21 features identified by Vidros et al. [5]. These features are placed into three categories: Linguistic, Contextual and Metadata. Linguistic features are related to the type of that the employer has used in the job description. Metadata features are related to the employer location, questions, logo etc. Table 1, lists all the 21 features used in this work. Predictive Model Building: We build OR Detector model using 3 base classifiers (J48, LR, and RF) and 3 ensemble techniques (AV, MV, and MXV). Using these, we create following 3 combinations:
Shloka Gilda presented concept approximately how NLP is relevant to stumble on fake information. They have used time period frequency-inverse record frequency (TF- IDF) of bi-grams and probabilistic context free grammar (PCFG) detection. They have examined their dataset over more than one class algorithms to find out the great model. They locate that TF-IDF of bi-grams fed right into a Stochastic Gradient Descent model identifies non-credible resources with an accuracy of seventy-seven.
B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli,( 2011) ―Bagging classifiers for fighting poisoning attacks in adversarial classification tasks.
Dros et al. (2016) determined the frauds exposed by job seekers through online recruitment services. They found that ORF is a new field of current severe vulnerable. Three current methods of fraud and scams in the online recruitment include Fake Job Advertisement, Economic trickery using fake job advertisements published on online recruitment boards, and reputed and real business enterprise publishing fake vacancy announcement.

### 2.2 Existing System

Employment scam is one of the serious issues in recent times addressed in the domain of Online Recruitment Frauds (ORF). In recent days, many companies prefer to post their vacancies online so that these can be accessed easily and timely by the job-seekers. However, this intention may be one type of scam by the fraud people because they offer employment to job-seekers in terms of taking money from them. Fraudulent job advertisements can be posted against a reputed company for violating their credibility. These fraudulent job post detection draws a good attention for obtaining an automated tool for identifying fake jobs and reporting them to people for avoiding application for such jobs.There are few applications that enables us to predict job detection of a single commodity with the use of supervised algorithms and the precision and accuracy of those models was not so high.

### 2.3 Proposed System

Machine learning approach is applied which employs several classification algorithms for recognizing fake posts. In this case, a classification tool isolates fake job posts from a larger set of job advertisements and alerts the user. To address the problem of identifying scams on job posting, supervised learning algorithm as classification techniques are considered initially. A classifier maps input variable to target classes by considering training data. Classifiers addressed in the paper for identifying fake job posts from the others are described briefly. These classifiers- based predictions may be broadly categorized into -Single Classifier based Prediction and Ensemble Classifiers based Prediction . The target of this study is to detect whether a job post is fraudulent or not. Identifying and eliminating these fake job advertisements will help the job-seekers to concentrate on legitimate job posts only. In this context, a dataset from Kaggle is employed that provides information regarding a job that may or may not be suspicious. The dataset has the schema
This dataset is used in the proposed methods for testing the overall performance of the approach. For better understanding of the target as a baseline, a multistep procedure is followed for obtaining a balanced dataset. Before fitting this data to any classifier, some pre-processing techniques are applied to this dataset. Pre-processing techniques include missing values removal, stop-words elimination, irrelevant attribute elimination and extra space not be sufficient enough. Here we use svm and random forest classifiers.

## 3. SYSTEM ANALYSIS

### 3.1 Functional Requirements Specifications

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly; the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based duplication systems by resorting to the hybrid cloud architecture.

- Data Collection
- Handling Missing Values
- Remove Duplicate rows
- Data Cleaning
- Finding Feature Importance
- Model Creating
- Model Training
- Hyper parameter tuning(Max Depth)
- Model Evaluation

### 3.2 Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Programming Language : Python
- Front end :Jupyter lab
- Tools : PIP
- Technology :Machine learning

### 3.3 Hardware Requirements

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum
- System Type : 64-bit Operating System

### 3.4 Performance requirements

Performance is measured in terms of the output provided by the application which is mostly based on accuracy. Requirement specification plays an important role in the analysing the system. Only when the requirement specifications are properly given, it is possible design a system. The requirement specification for any system can be stated as follows:

1. The system should be precise.
2. The system should be able to interface with the existing system.
3. The system should be better than the previous system.

## 4. SYSTEM DESIGN

### 4.1 Architecture of Proposed System

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified necessities. It might be viewed as a product development application of systems theory. This project is being developed in Python, and the technology involved is MACHINE LEARNING.

The architecture of the proposed system involves the input data, the trained model and the output. The user provides the feature values as input and these are processed by the trained model. Further the output is shown to the user via the interface. Training and further proceed with performance evaluation. Hence, we achieve the resulting in trained model. We can provide the input and then predict the output.

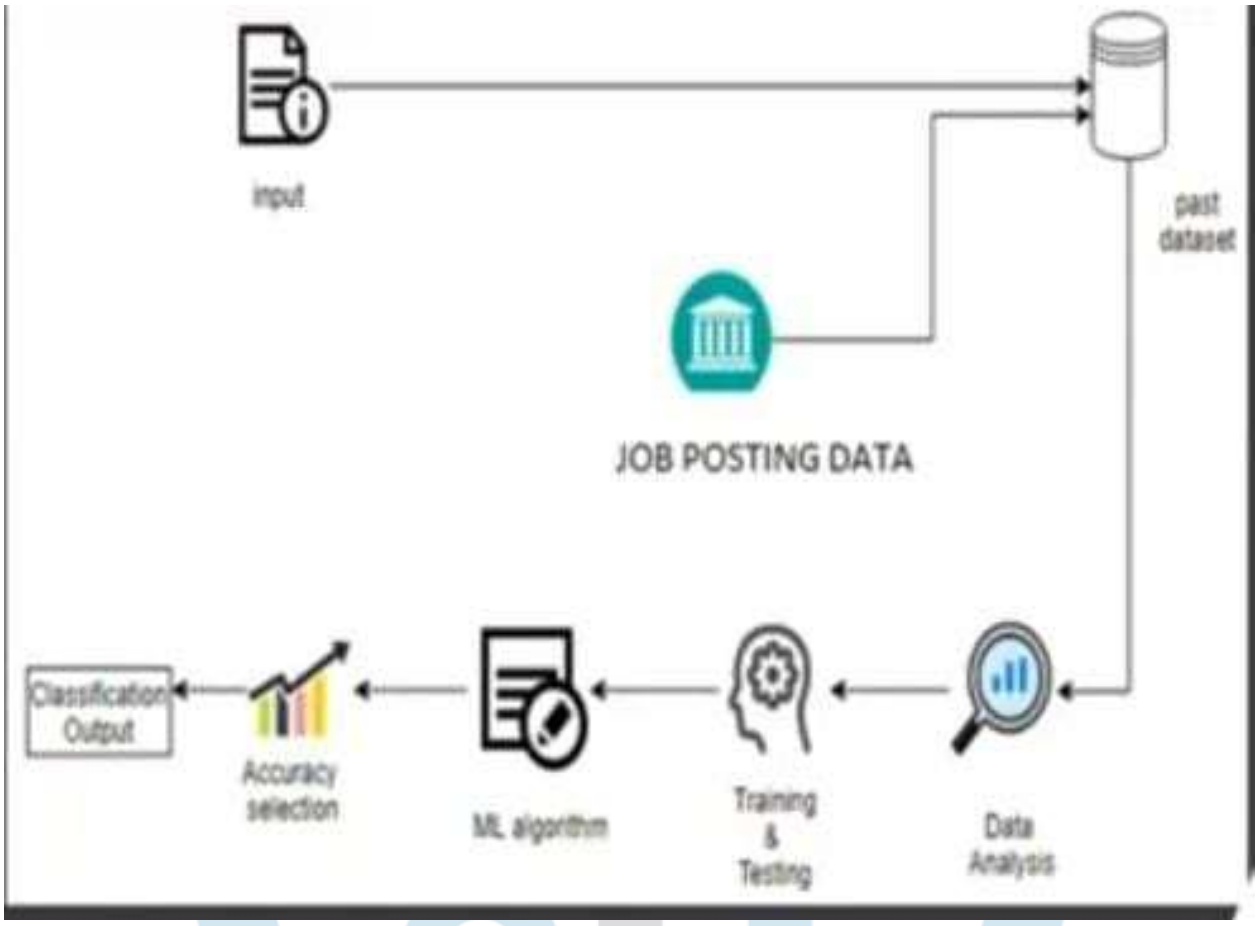



**Fig 4.1 : Architecture Diagram**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**4.2 Modules**

Weather holds a significant role in our daily life, where we can use weather forecasts to avoid future disasters or any other hazardous events. Several techniques were proposed by researchers to aid in such tasks.

**4.2.1. Modules Name**

1. Data set
2. Understanding the data sheets
3. Pre- processing
4. Splitting data
5. Model building
6. Evaluation

**4.2.2. Dataset Module:**

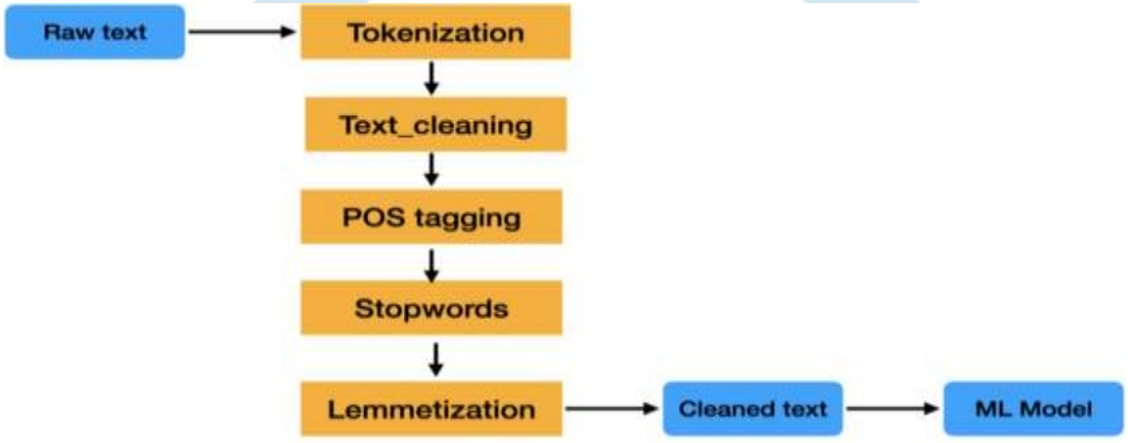
The target of this study is to detect whether a job post is fraudulent or not. Identifying and eliminating these fake job advertisements will help the jobseekers to concentrate on legitimate job posts only. In this context, a dataset from Kaggle is employed that provides information regarding a job that may or may not be suspicious.

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

**4.2.3. Understanding the dataset Module:**
Once the dataset collection is done, we need to analyze and visualize the data. The data is accessed by using pandas and visualization is carried by matplotlib. Plotting of the data is done for attributes such as loan status, history credit. Visualization gives a clear idea about the information in visual content.

**4.2.4. Pre processing module:**
This dataset contains 17,880 number of job posts. This dataset is used in the proposed methods for testing the overall performance of the approach. For better under -standing of the target as a baseline, a multistep procedure is followed for obtaining a balanced dataset. Before fitting this data to any classifier, some pre-processing techniques are applied to this dataset. Pre-processing techniques include missing values removal, stop-words elimination, irrelevant attribute elimination and extra space.
NLP stands for Natural Language Processing that automatically manipulates the natural language, like speech and text in apps and software. Speech can be anything like text that the algorithms take as the input, measures the accuracy, runs it through self and semi-supervised models, and gives us the output that we are looking forward to either in speech or text after input data.

NLP is one of the most sought-after techniques that makes communication easier between humans and computers. If you use windows, there is Microsoft Cortana for you, and if you use macOS, Siri is your virtual assistant. The best part is even the search engine comes with a virtual assistant. Example: Google Search Engine.



Steps in NLP

**4.3 UML Diagrams**
The UML diagrams helps in better visualization of any project. They help in understanding the project and also describe the purpose with easy visualization.
Some of the important uses of UML diagrams is described as follows:

- It helps in understanding the behaviour of the end resultant system.
- It helps in communicating the proposed project effectively with no misconceptions.
- It helps in getting to know the requirements and also helps to ideate from where we can implementation i.e., the beginning step.

In the following section, we have discussed about the different UML diagrams for our project.

**4.3.1. Use Case Diagram**
The use case diagram gives the details regarding the actors and their functionalities. It is basically a graphical representation.
Following use case diagram consists of the following actors.

- User
- User interface
- Trained model

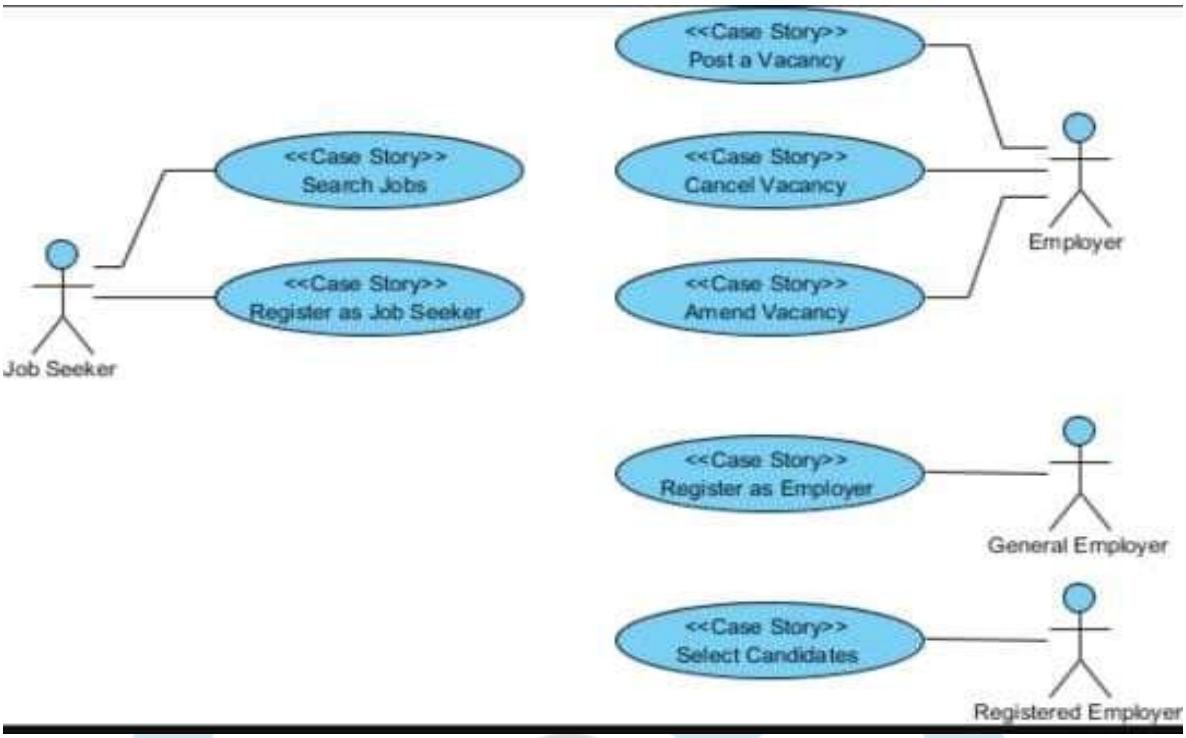



**Fig 4.2.1 : Use Case Diagram**

**4.3.2. Class Diagram**

The class diagram plays a prominent role in object-oriented programming. The class diagram depicts the static structure which gives information about the classes in the system, the attributes and operations as well as the relationship amongst the various classes. The class diagram provides details about various classes in the system.

Following is the class diagram describing the different classes containing attributes and methods in our system.

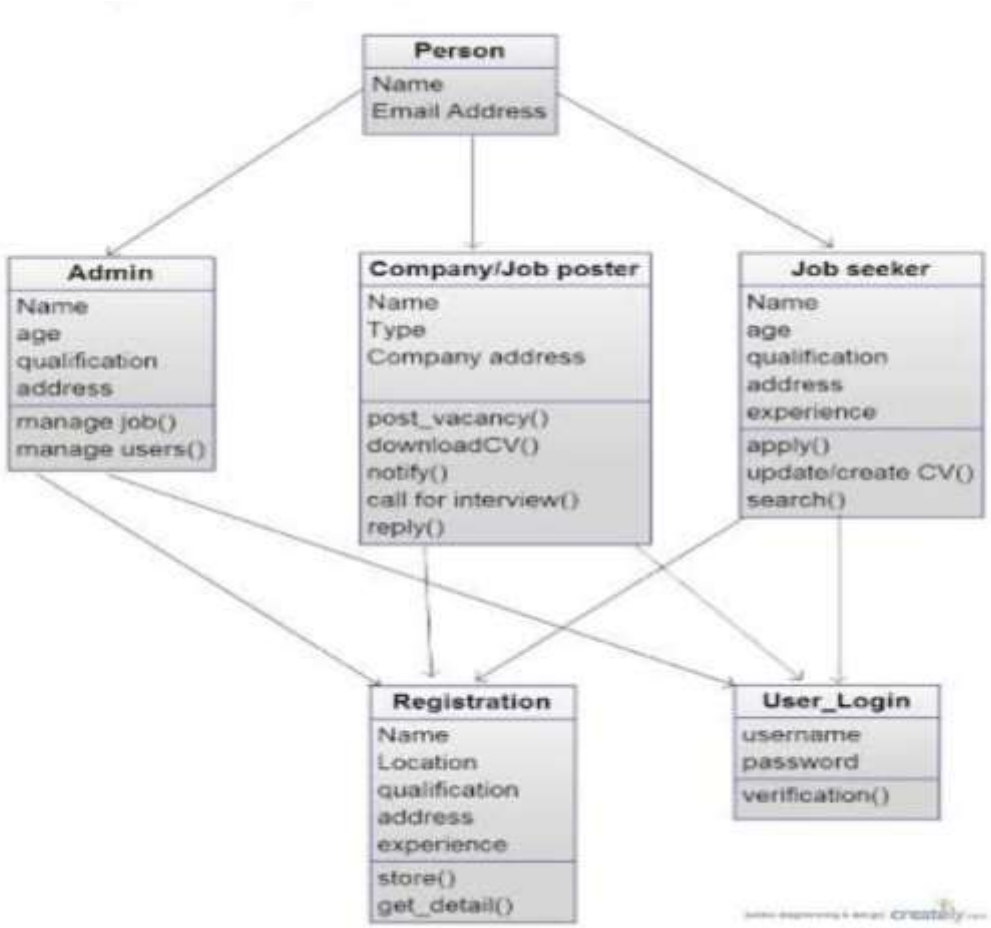



**Fig 4.2.2 : Class Diagram**

**4.3.3. Sequence Diagram**

The sequence diagram in UML provides information about how the operations are carried out. In other words, they provide details on what messages are sent and when. It is an interaction diagram where details are organised according to time. It provides dynamic view of the system i.e., the model.

Following is the sequence diagram for our model:

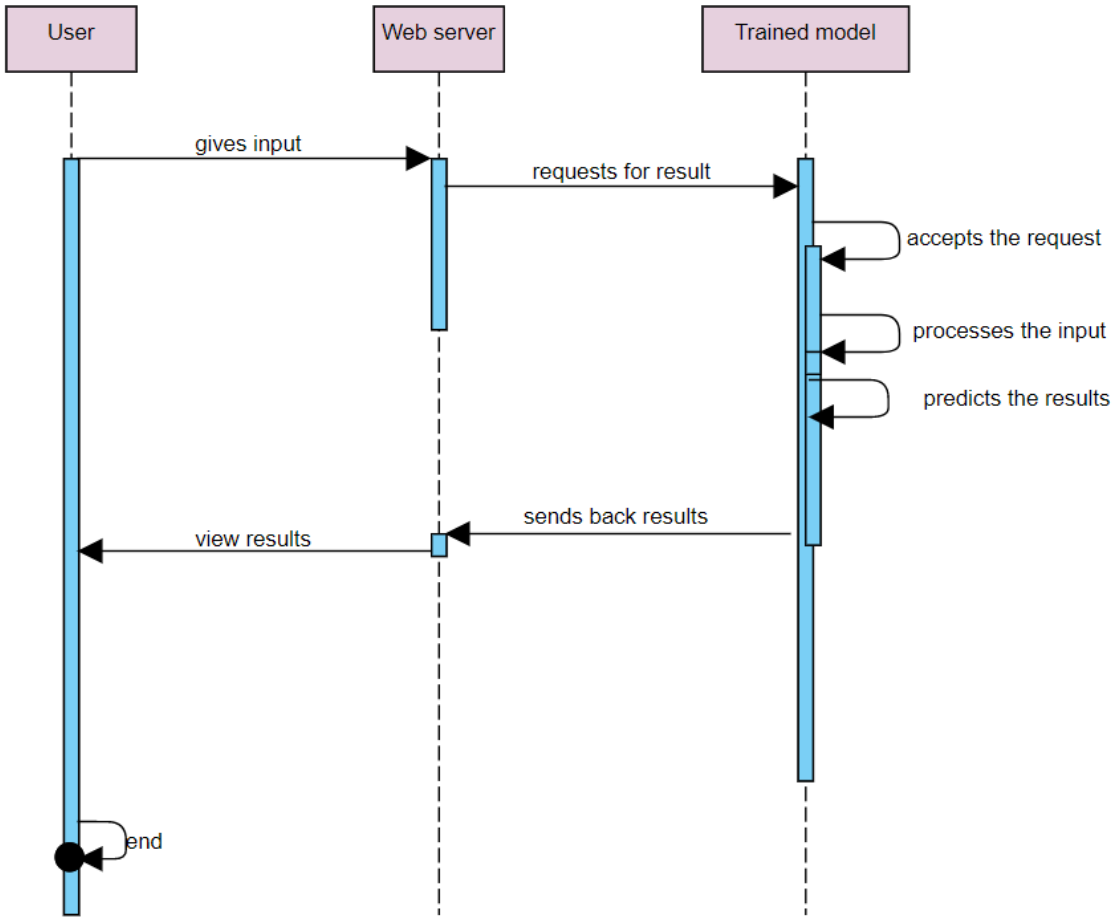



**Fig 4.2.3 : Sequence Diagram**

**4.3.4. Activity Diagram**

The activity diagram is used for representing the work flow of the activities. It helps in showing the stepwise activities for a model. It shows the control flow from the beginning point until the finishing point depicting various decision paths that are present when the activity is executed.

The activity diagram consists of rounded rectangles which are used for representing actions, bars which are used for representing the start or end of the activities and a black circle depicting the start and end of the control flow.

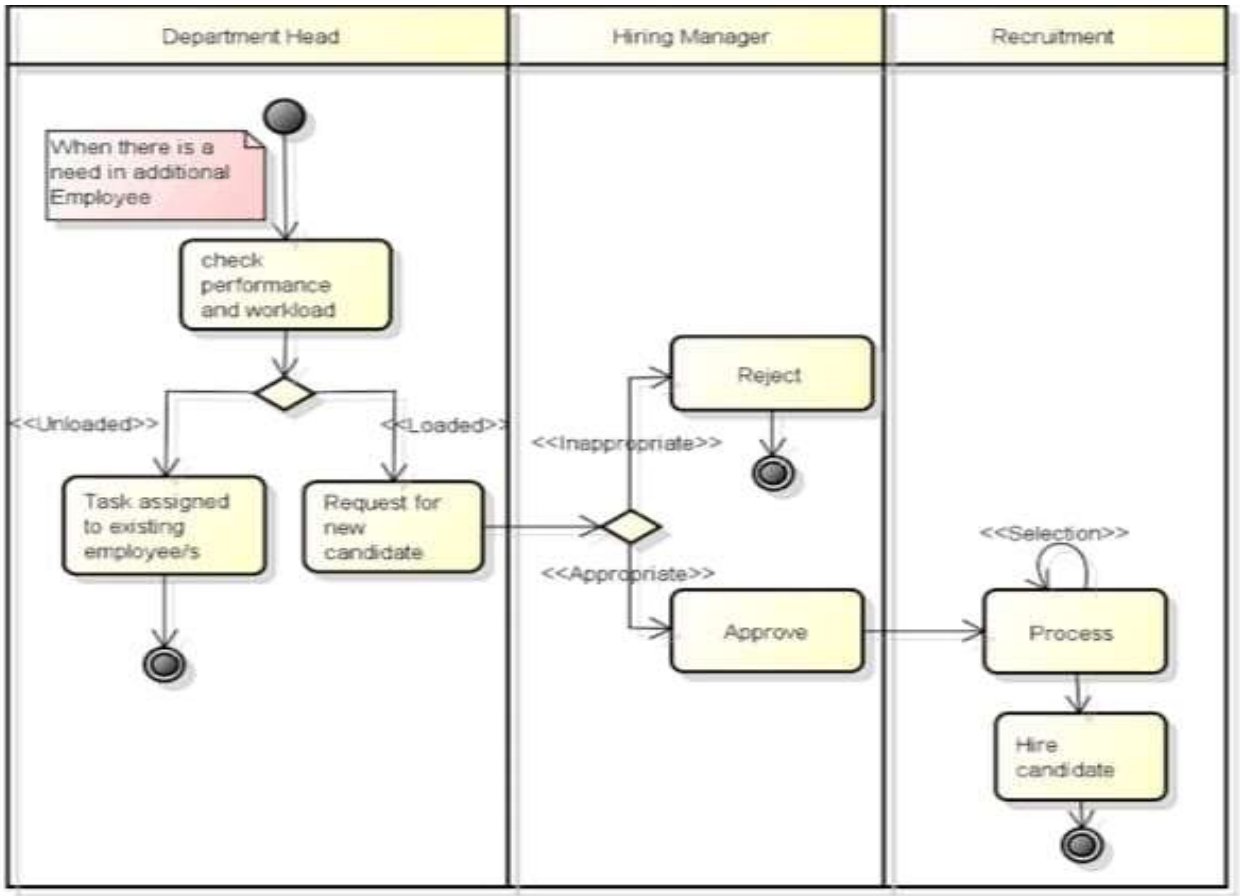



**Fig 4.2.4 :** Activity Diagram

**4.3.5. State chart Diagram**

State chart diagrams define different states of an object during its lifetime and these states are changed by events. Modeling reactive systems with state chart diagrams is useful. The term "reactive system" refers to a system that reacts to external or internal events. They describe how control moves from one state to another. A state is defined as a state in which an object exists and changes when an event occurs. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

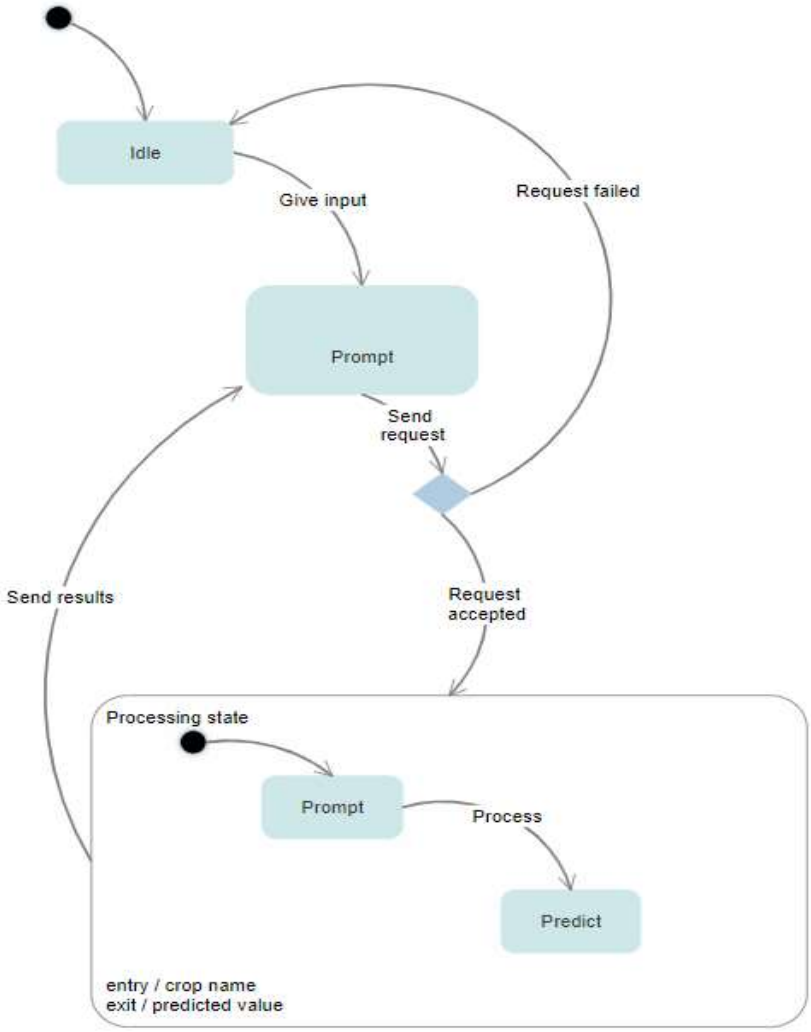



**Fig: 4.2.5** State chart Diagram

**4.3.6. Collaboration Diagram**

The collaborations are used when it is essential to depict the relationship between the object. The sequence and collaboration diagrams both depict the same information, but they do so in quite different ways. Use cases are best analyzed using collaboration diagrams.

Following are some of the use cases enlisted below for which the collaboration diagram is implemented:

- To represent collaboration between the objects or roles that contain the use cases and operations' functionalities.
- To model the mechanism within the system's architectural design.
- To document the interactions that represent the flow of information between the items and roles inside the collaboration.
- To model different scenarios within the use case or operation, involving a collaboration of several objects and interactions.
- To support the identification of objects participating in the use case.
- In the collaboration diagram, each message constitutes a sequence number, such that the top-level message is marked as one and so on.
- The messages transmitted during the same call have the same decimal prefix, but various suffixes of 1, 2, and so on, depending on their frequency.

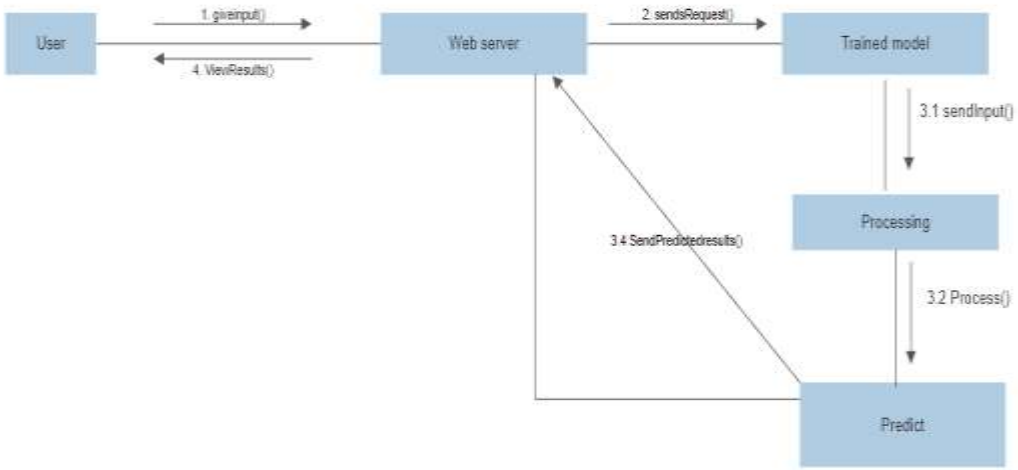



**Fig: 4.2.6** Collaboration Diagram

## 5. IMPLEMENTATION

### 5.1 Methodology

Any project's implementation stage is a true representation of the defining moments that determine whether a project succeeds or fails. The system or system modifications are installed and made operational in a production environment during the implementation stage. After the system has been tested and accepted by the user, the phase begins. This phase continues until the system meets the established user requirements and is ready to go into production. Machine learning primarily comprises of three learning approaches for training a model: supervised learning, reinforcement learning, and unsupervised learning. Supervised learning is a type of learning that maps known inputs into outputs that are mapped from input to output. Here, we are using Supervised Machine learning algorithm to achieve our task

#### 5.1.2 Data Pre-processing

The most important part of every machine learning based system is data pre-processing. The dataset obtained maybe in raw format and cannot be fed directly into the machine. Almost all the datasets must be cleaned, processed and transformed in order to make them more precise and usable. This is because any un processed dataset consists of missing values, duplicates, outliers and null values etc. Such dataset when used without processing leads to less accurate results which are disastrous. Generally, every dataset is different and hence we can face various challenges. There maybe different kinds of data which has to be made homogeneous accordingly with respect to the technologies used. In the case of our model, the missing values, duplicate values and the null values are removed in the data pre-processing. Outliers are also eliminated.

- Data standardization
- Data Cleaning
    - Handling missing data
    - Handling null values
    - Handing duplicate values
    - Managing the outliers

Data standardization**:**

Data standardization is also a vital data pre-processing technique as it helps in eliminating or removing the inconsistent data. The process of data standardization helps in making the consistent internally. It is handy in transforming attributes acquiring a Gaussian distribution which differ means standard deviation into a Gaussian Standard Gaussian Distribution. In our project we standard our

data in the dataset using the scikit-learn library.

Data cleaning**:**

Data cleaning is one in all necessary steps in the attaining good accuracy in machine learning. It is also an essential step in our project. Data cleaning, although having great importance is mostly a hidden step that is not mentioned by most of the professionals. It helps in uncovering and cleaning the missing values, outliers the inconsistent data which may interrupt in successful execution of the model. In general, if we are using a better data which is clean there is a chance of better results even when we use simple algorithm. Our dataset contains different attributes of different types. Hence it would require different methods for cleaning the dataset.

The data cleaning techniques we used in our project are elaborated as follows:

- Handling missing data: Missing data could be misleadingly tough issue in the field of machine learning. These may lead to miss interpretation and wrong results. We cannot simply remove the record in order to overcome this issue. We need to identify a method that would helping in filling up these missing values so that the important records could not be missed. We deal this issue in two ways. The former is by removing the entry of the missing values and the latter is by filling it up based on the past observations. The former step can be performed if the data is less quality or unimportant. This method may not be recommended as the information which is important would be lost. The latter is inserting the missing values using appropriate statistical methods. We can inert values in the missing columns by finding out the mean and then filling the column.
- Handling null values: The values also possess the same issue as the missing values possess the null values can be avoided by simply deleting the record or filling up the entry by appropriate statistical method. In our project, we used the mean of the attribute column or the previous entries in or to deal with null values. We used an appropriate method available from the library available in python to perform the statistical operations.
- Removing the duplicates: The presence of the duplicate entries the dataset would have a significant affect on our model or any other machine learning model. Hence the issue of duplicates must be resolved in order to successful execution of the model. In our project, we used the drop duplicates method in order to remove the duplicates in our dataset. This method is elaborate in the upcoming sections.
- Managing the outliers: The outliers also possess a challenge in developing a machine learning model. For instance, we take the presence of outliers in a particular model, the linear regression could be less fast or accurate than the decision-tree. It is not reasonable to have unlikely data or unreasonable data in the dataset. For example, we cannot have age of 300 entry in our dataset as it would be practically impossible and also has a negative effect to our model. Hence it is a better idea or approach to remove the outliers and work on the project. For our project, we plotted different graphs on the attributes in order to identify and detect the outliers for the attributes. Significantly we also transformed and modified the data for improving our dataset. Hence, we applied the methods available in the python library to study the data in the dataset so that we could identify any abnormal entry or anomalies in the data. Python libraries used have provided us with appropriate methods to visualize the data and detect the missing values and outliers and overcome the problems in our dataset.

### 5.1.3 Language Used for Implementation

For the implementation of the system, we used jupyter notebook to write the python code for the system. Jupyter notebook is an open-source web- application which helps programmers to edit and create code with less effort and in less time. The jupyter notebook notebook supports various programmers to use python more efficiently in big projects. It supports various libraries Pandas, Numpy, Sklearn and many more libraries. One of the major advantages of jupyter notebook is that we can execute code block by block. This makes the debugging process easier and also better handle the errors.Also we use HTML,CSS.

Some of the libraries used in our system are Pandas, Numpy, sklearn and Flask. The provide various advantages for the process of machine learning. For instance, the Pandas library provides easy to use data structures with high performance and data analysis tool for the Python Programming Language.

**SPACY**

Spacy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. Spacy is designed specifically for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for Machine learning and deep learning.

**WORD CLOUD**

Word cloud - Many times you might have seen a cloud filled with lots of words in different sizes, which represent the frequency or the importance of each word. This is called Tag Cloud or Word Cloud. For this tutorial, you will learn how to create a Word Cloud of your own in Python and customize it as you see fit. This tool will be quite handy for exploring text data and making your report easily.

**Numerous Areas of Word Cloud Usage**

**Top topics on social media:**

If we could read and get text of posts/tweets that users are sending out, we can extract the top words out of them and they could be used in the trending section to classify and organise posts/tweets under respective sections.

**Trending News Topics:**

If we can analyse the text or headings of various news articles, we can extract the top words out of them and identify what are the most trending news topics around a city, country or the whole world.

**Navigation systems for Websites:**

Whenever you visit a website that is driven by categories or tags, a word cloud can actually be created and the users can directly jump to any topic while knowing the relevance of the topic across the community.

## 5.2 Implementation

### 5.2.1 Exploratory Data Analysis

This is an important and more interesting step in the entire process. Exploratory Data Analysis is a typical design thinking step. In this step, we study the data in depth in order to identify hidden insights in it. For this reason, it is said to be brainstorming stage of Machine Learning. In this stage, we use various data visualization techniques in the form graphs and plots in order to gain insights and various correlations between the variables. This may help in detecting any data imbalances and also reduce the data for making it more efficient. We use the library functions available in python to fulfil this purpose.

### 5.2.2 Algorithms Implemented

The main purpose of the project is to identify an algorithm which provides better accuracy than the existing system. In this case, we implement different machine learning models and identify the algorithm which has the highest accuracy. The performance evaluation is then performed on the model and then accordingly we build a concluding trained model. The model is then deployed to predict prices using the algorithm. Following is the algorithm we implemented in our system.

**SVM(SUPPORT VECTOR MACHINE):**

‘‘Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

While evaluating the performance skill of a model, it is necessary to employ some metrics to justify the evaluation. For this purpose, following metrics are taken into consideration to identify the best relevant problem-solving approach. Accuracy is a metric that identifies the ratio of true predictions over the total number of instances considered. However, the accuracy may not be enough metric to consider wrong predicted cases. If a fake post is treated as a true one, it creates a significant problem. Hence, it is necessary to consider false positive and false negative cases that compensate to misclassification.

For measuring this compensation, precision and recall is quite necessary to be considered . Precision identifies the ratio of correct positive results over the number of positive results predicted by the classifier. Recall denotes the number of correct positive results divided by the number of all relevant samples. F1-Score or F-measure is a parameter that is concerned for both recall and precision and it is calculated as the harmonic mean of precision and recall.

confusion matrix which processes the output by a certain parameter such as true positive, true negative, false positive and false negative in matrix form and gives an evaluation result evaluating model ‘s performance since it does not consider wrong predicted cases. If a fake post istreated as a true one, it creates a significant problem. Hence, it is necessary to consider false positive and false negative cases that compensate to misclassification.

For measuring this compensation, precision and recall is quite necessary to be considered [7]. Precision [14] identifies the ratio of correct positive results over the number of positive results predicted by the classifier. Recall [14] denotes the number of correct positive results divided by the number of all relevant samples. F1-Score or F-measure [14] is a parameter that is concerned for both recall and precision and it is calculated as the harmonic mean of precision and recall

- **Accuracy = Accuracy (all correct / all) = (TP + TN) /( TP + TN + FP + FN)**
- **Precision (true positives / predicted positives) = TP / (TP + FP)**
- **Sensitivity aka Recall (true positives / all actual positives) = TP / (TP + FN)**
- **Specificity (true negatives / all actual negatives) =TN / (TN + FP)**

The default values for the parameter controlling the size of the tree is max_depth.to reduce memory consumption and complexity and size of the trees max_depth parameter is used.

The performance of decision tree regression is measured by three equation they are:-
a. **Mean absolute error** :The **Mean Absolute Error**(MAE) is the average of all absolute errors. The formula is:

$$\mathbf{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{x}|$$

**Where**:

- n = the number of errors,
- Σ = summation symbol (which means "add them all up"),
- $|x_i - x|$ = the absolute errors.

b. **Mean squared error :** The **mean squared error** (MSE) tells you how close a regression line is to a set of points. It accomplishes this by squaring the distances between the points and the regression line (these distances are the "errors"). Squaring is required to eliminate any negative signals. It also gives significant discrepancies greater weight. Because you're calculating the average of a series of errors, it's termed the mean squared error. The better the forecast, the lower the MSE.

c. **R mean square score:** It is the square root of the mean of all the errors squared. For numerical predictions, the root mean square error (RMSE) is regarded as an effective general-purpose error metric. Because RMSE is scale-dependent, it should only be used to evaluate prediction errors of different models or model configurations for a single variable, not between variables.

### 5.3 Result

In our project, we have tried to determine an algorithm having accuracy greater than the existing system. The data pre-processing helps in furnishing the dataset and the exploratory data analysis helps in gaining better insights. After implementing the dataset on the machine learning model, we determine the algorithm with highest accuracy as the winning model. The accuracy rates for the different models are shown in the following table:

**Classification Report:** A Classification report is used to measure the quality of predictions from a classification algorithm. The below report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **0** | **0.96** | **1.00** | **0.98** | **5093** |
| **1** | **0.98** | **0.31** | **0.47** | **271** |
| **Accuracy** | | | **0.96** | **5364** |
| **Macro avg** | **0.97** | **0.65** | **0.73** | **5364** |

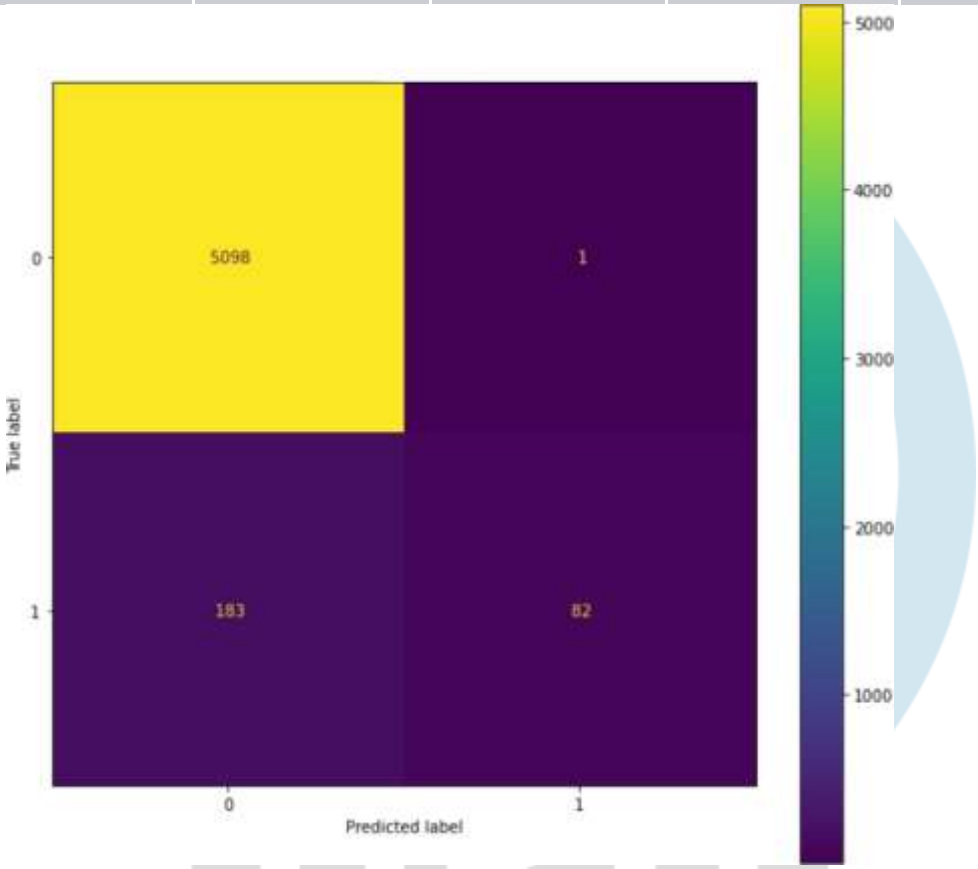
In our Project we got accuracy score by calculating using above formula.

**Accuracy Score:** 0.9701715137956749

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a

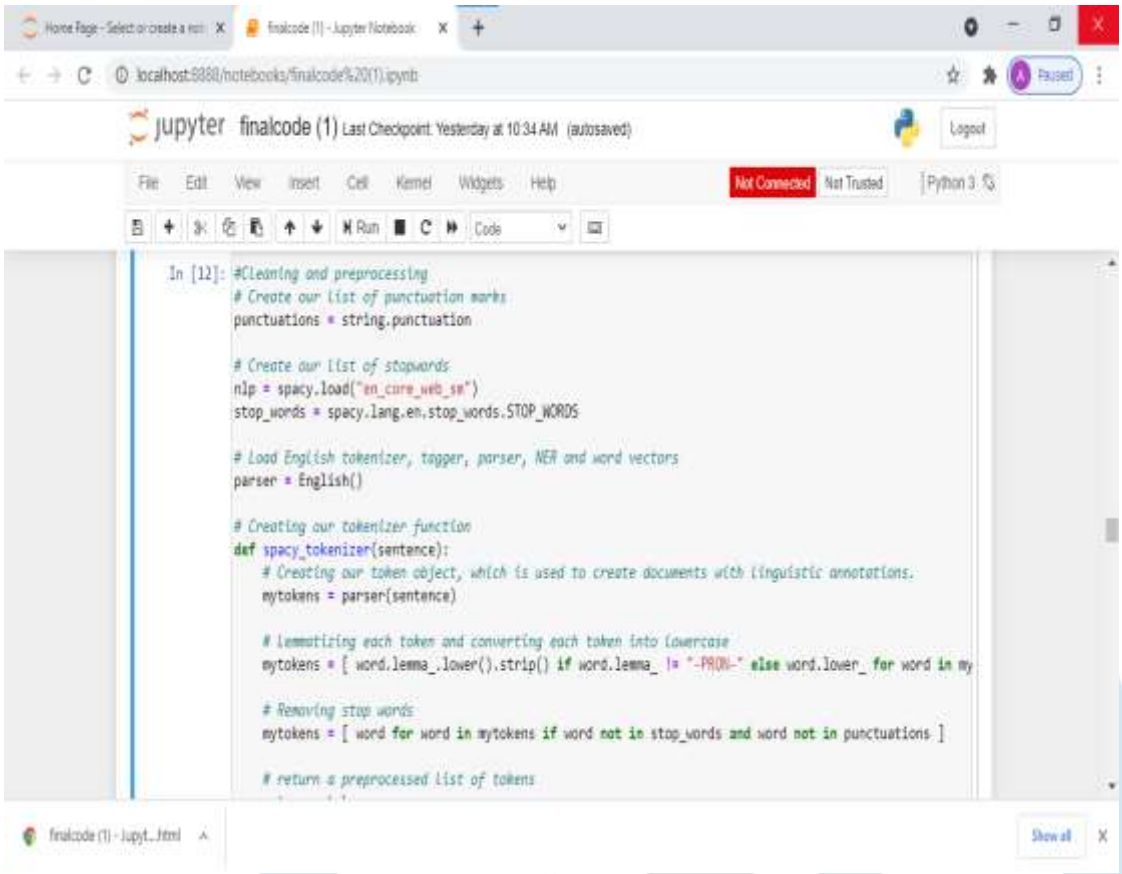
| | | | | |
|---|---|---|---|---|
| **Weighted avg** | **0.97** | **0.96** | **0.96** | **5364** |

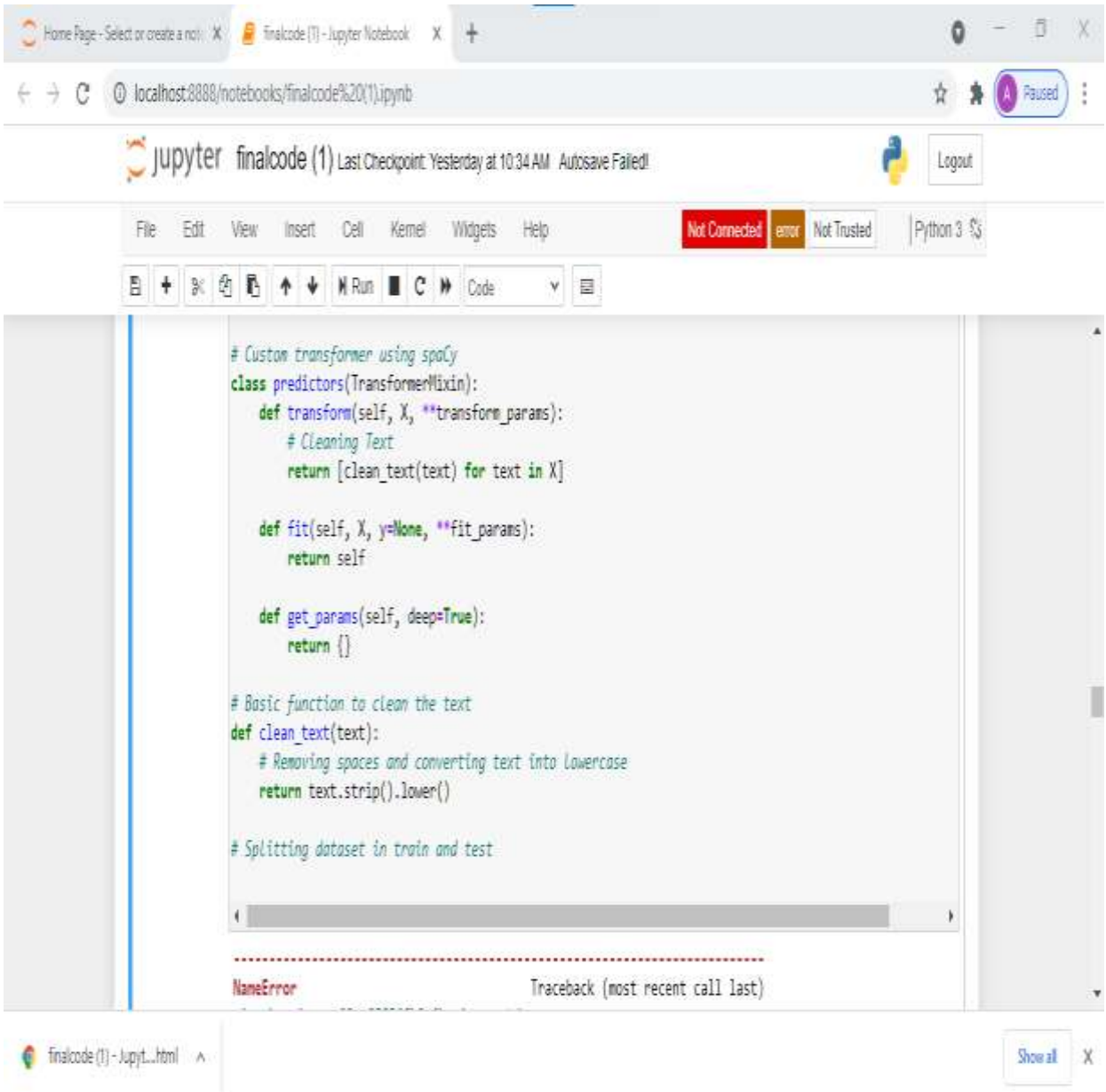
set of test data for




which the true values are known.
**"True positive"** for correctly predicted event values. **"False positive"** for incorrectly predicted event values. **"True negative"** for correctly predicted no-event values. **"False negative"** for incorrectly predicted no-event values. **In below Confusion matrix the we got**
True positive=5098

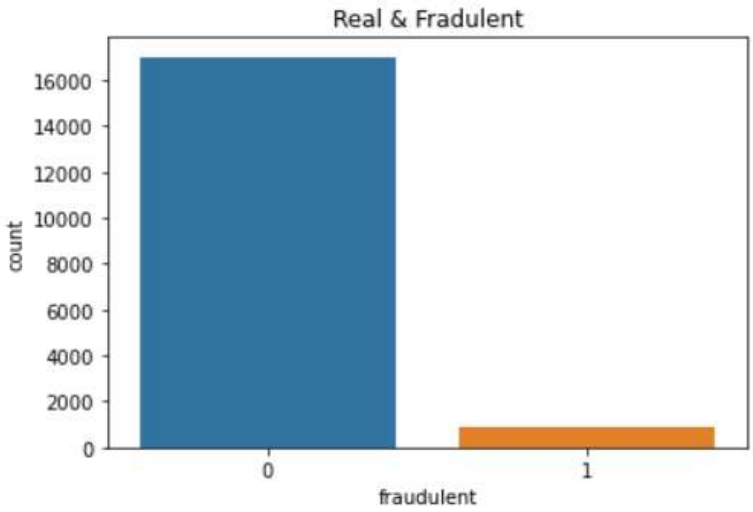
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$


jupyter finalcode (1) Last Checkpoint: Yesterday at 10:34 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Not Trusted
Python 3
Run
Code
In [12]:
#Cleaning and preprocessing
# Create our list of punctuation marks
punctuations = string.punctuation
# Create our list of stopwords
nlp = spacy.load("en_core_web_sm")
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 my
# Removing stop words
mytokens = [ word for word in mytokens if word not in stop_words and word not in punctuations ]
# return a preprocessed list of tokens
finalcode (1) - Jupyt...html
Show all


Home Page - Select or create a noti
finalcode (1) - Jupyter Notebook
localhost:8888/notebooks/finalcode%20(1).ipynb
Paused
jupyter finalcode (1) Last Checkpoint: Yesterday at 10:34 AM Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected error Not Trusted
Python 3
Run
Code
# 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()
# Splitting dataset in train and test
NameError Traceback (most recent call last)
finalcode (1) - Jupyt...html
Show all

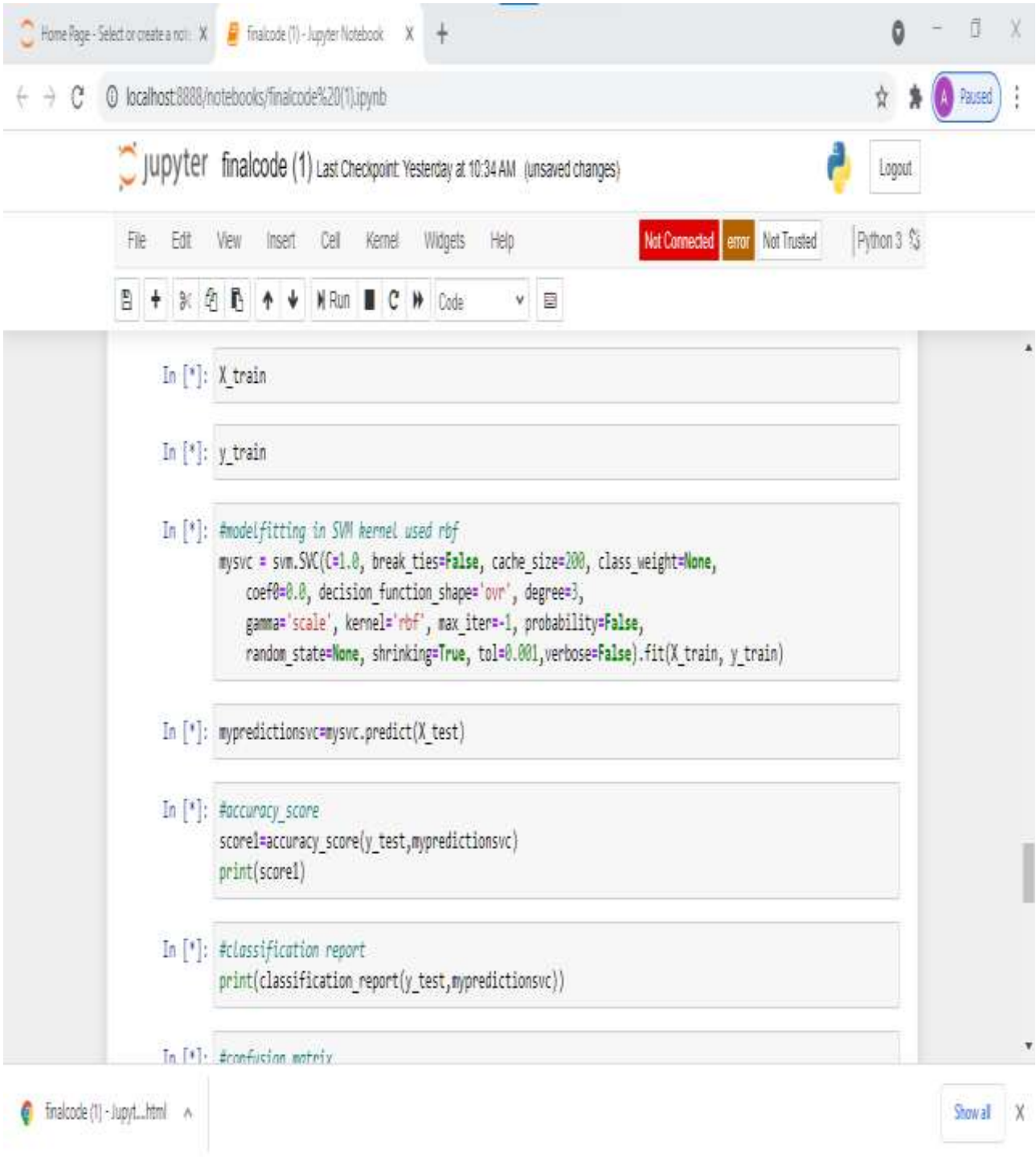
:

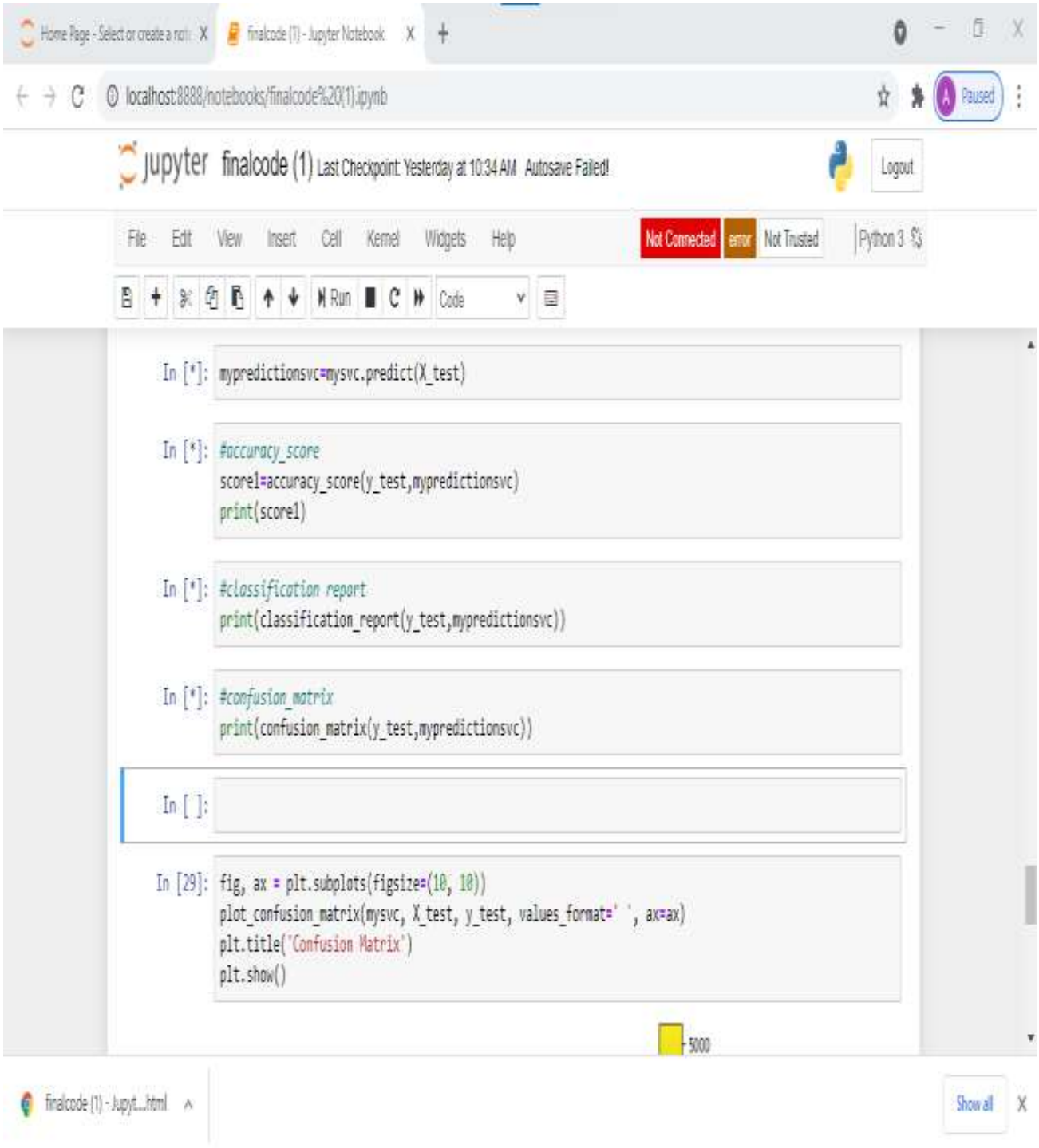
| | fraudulent | title |
|---|---|---|
| 0 | 0 | 17014 |
| 1 | 1 | 866 |

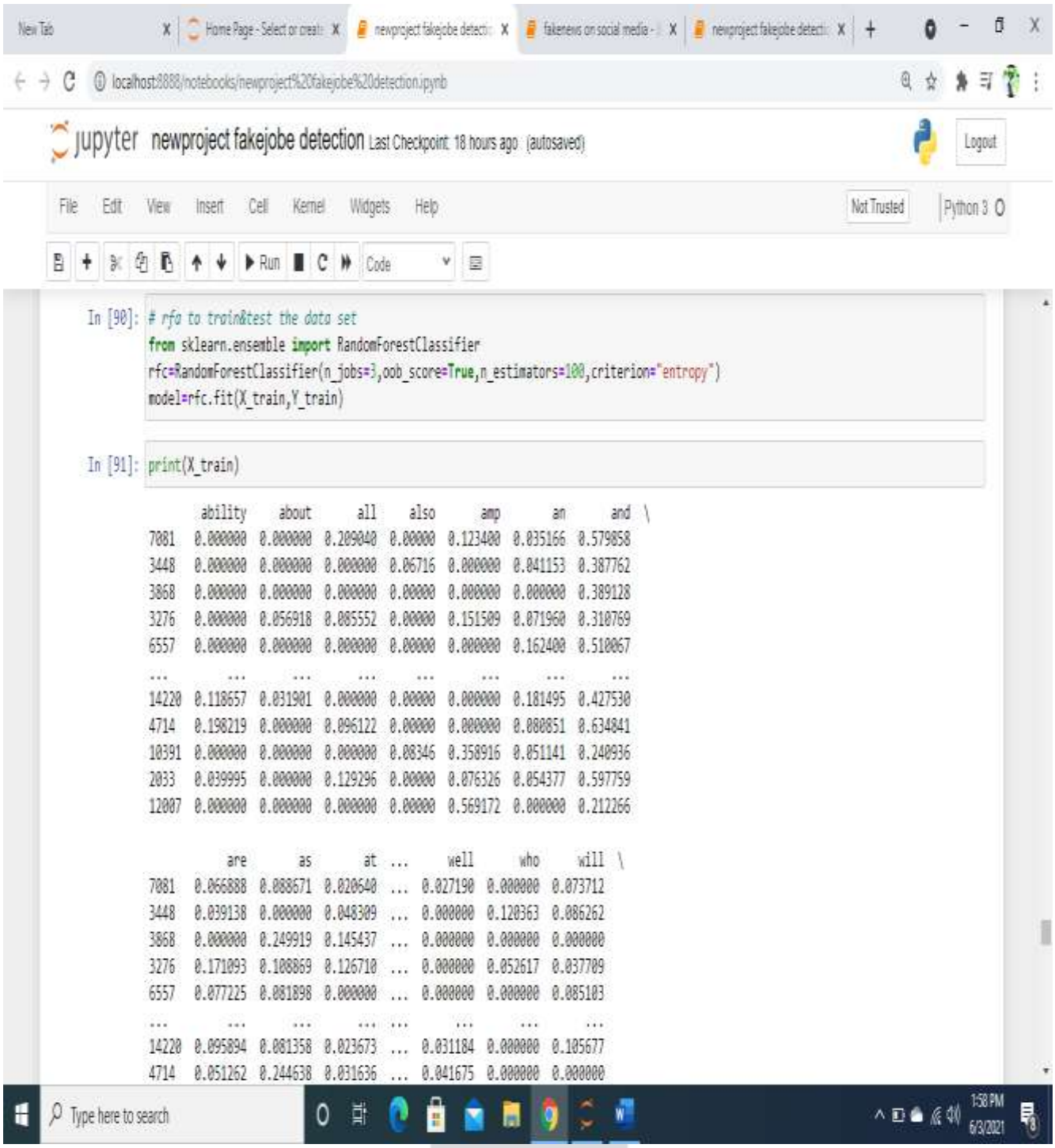


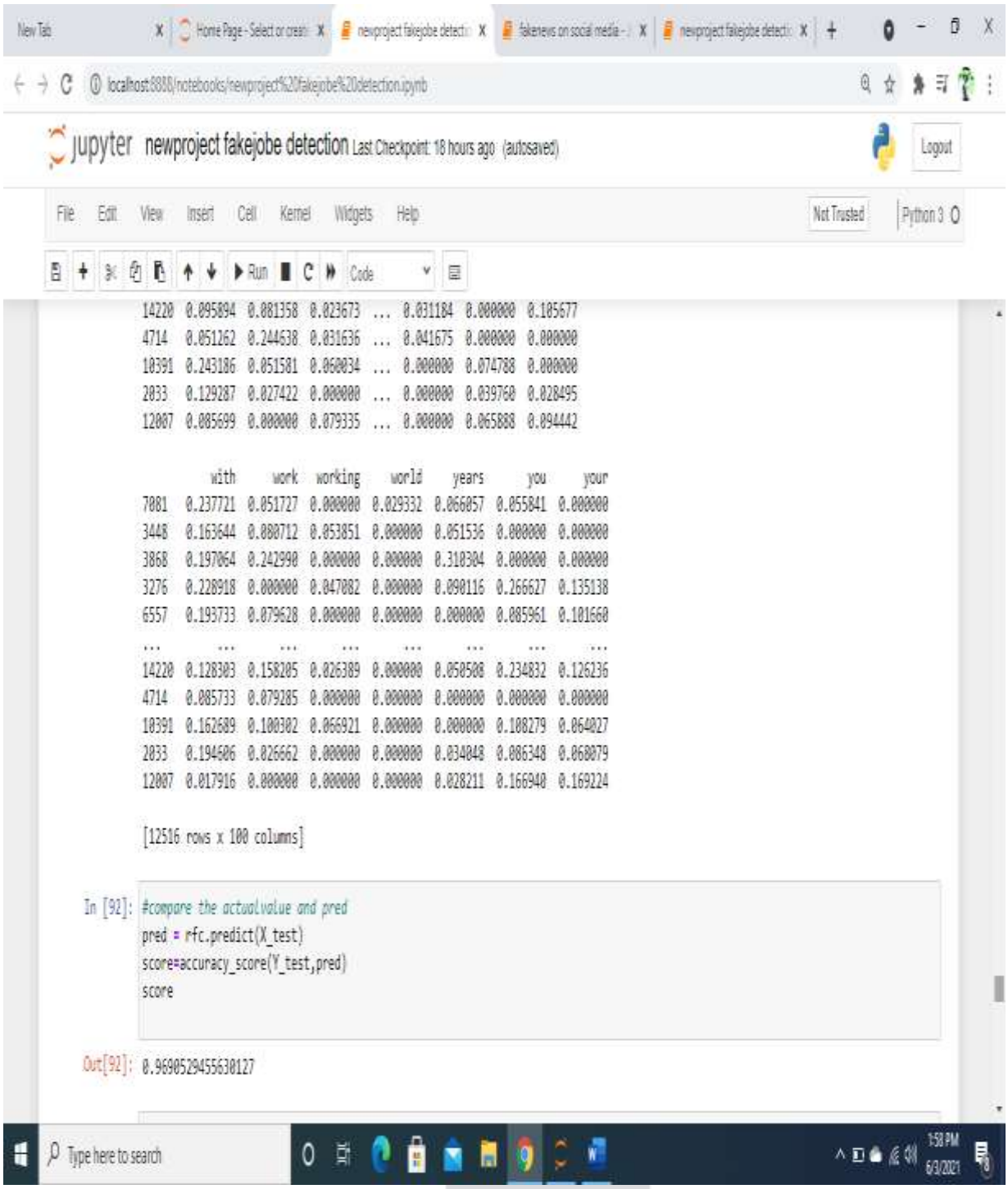

```
#Fraud and Real visualization
sns.countplot(data.fraudulent).set_title('Real & Fradulent')
data.groupby('fraudulent').count()['title'].reset_index().sort_values(by='title',ascending=False)
```

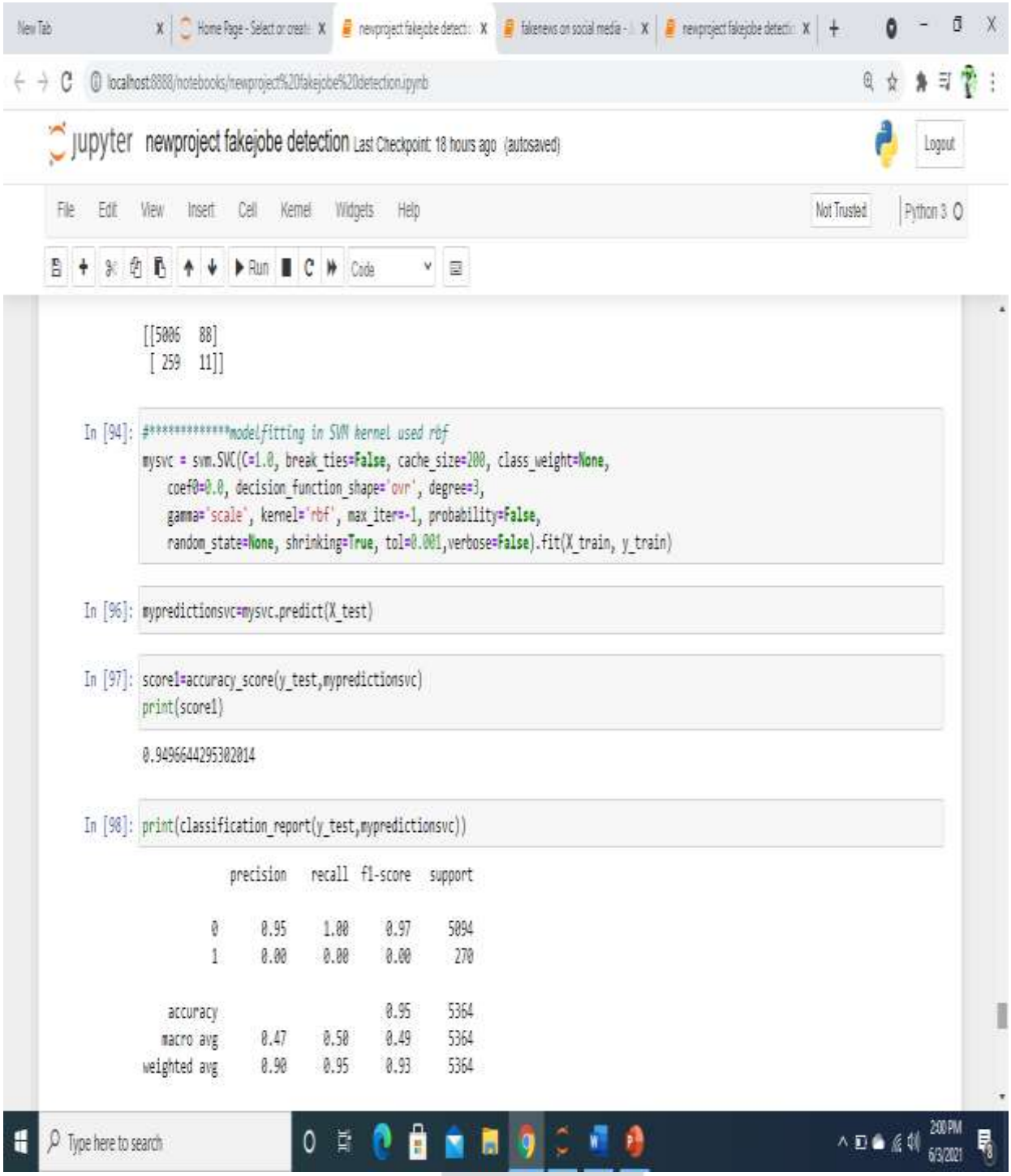
```
#combine text in a single column to start cleaning our data
data['text']=data['title']+' '+data['location']+' '+data['company_profile']+' '+data['description']+' '+data['requirements']
del data['title']
del data['location']
del data['department']
del data['company_profile']
del data['description']
del data['requirements']
del data['benefits']
del data['required_experience']
del data['required_education']
del data['industry']
del data['function']
del data['country']
```


Home Page - Select or create a noti
finalcode (1) - Jupyter Notebook
localhost:8888/notebooks/finalcode%20(1).ipynb
Paused
jupyter finalcode (1) Last Checkpoint: Yesterday at 10:34 AM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected error Not Trusted
Python 3
Run
Code
In [*]: X_train
In [*]: y_train
In [*]: #modelfitting in SVM kernel used rbf
mysvc = svm.SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0, decision_function_shape='ovr', degree=3,
gamma='scale', kernel='rbf', max_iter=-1, probability=False,
random_state=None, shrinking=True, tol=0.001,verbose=False).fit(X_train, y_train)
In [*]: mypredictionsvc=mysvc.predict(X_test)
In [*]: #accuracy_score
score1=accuracy_score(y_test,mypredictionsvc)
print(score1)
In [*]: #classification report
print(classification_report(y_test,mypredictionsvc))
finalcode (1) - Jupyt...html
Show all


Home Page - Select or create a noti
finalcode (1) - Jupyter Notebook
localhost:8888/notebooks/finalcode%20(1).ipynb
Paused
jupyter finalcode (1) Last Checkpoint: Yesterday at 10:34 AM Autosave Failed!
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected error Not Trusted
Python 3
Run
Code
In [*]: mypredictionsvc=mysvc.predict(X_test)
In [*]: #accuracy_score
score1=accuracy_score(y_test,mypredictionsvc)
print(score1)
In [*]: #classification report
print(classification_report(y_test,mypredictionsvc))
In [*]: #confusion_matrix
print(confusion_matrix(y_test,mypredictionsvc))
In [ ]:
In [29]: fig, ax = plt.subplots(figsize=(10, 10))
plot_confusion_matrix(mysvc, X_test, y_test, values_format=' ', ax=ax)
plt.title('Confusion Matrix')
plt.show()
5000
finalcode (1) - Jupyt...html
Show all


New Tab
Home Page - Select or creat
newproject fakejobe detecti
fakenews on social media -
newproject fakejobe detecti
localhost:8888/notebooks/newproject%20fakejobe%20detection.ipynb
jupyter newproject fakejobe detection Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [90]:
# rfa to train&test the data set
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_jobs=3,oob_score=True,n_estimators=100,criterion="entropy")
model=rfc.fit(X_train,Y_train)
In [91]:
print(X_train)
ability about all also amp an and \
7081 0.000000 0.000000 0.209040 0.00000 0.123400 0.035166 0.579858
3448 0.000000 0.000000 0.000000 0.06716 0.000000 0.041153 0.387762
3868 0.000000 0.000000 0.000000 0.00000 0.000000 0.000000 0.389128
3276 0.000000 0.056918 0.085552 0.00000 0.151509 0.071960 0.310769
6557 0.000000 0.000000 0.000000 0.00000 0.000000 0.162400 0.510067
... ... ... ... ... ... ... ...
14220 0.118657 0.031901 0.000000 0.00000 0.000000 0.101495 0.427530
4714 0.198219 0.000000 0.096122 0.00000 0.000000 0.000851 0.634841
10391 0.000000 0.000000 0.000000 0.08346 0.358916 0.051141 0.240936
2033 0.039995 0.000000 0.129296 0.00000 0.076326 0.054377 0.597759
12007 0.000000 0.000000 0.000000 0.00000 0.569172 0.000000 0.212266
are as at ... well who will \
7081 0.066888 0.088671 0.020640 ... 0.027190 0.000000 0.073712
3448 0.039138 0.000000 0.048309 ... 0.000000 0.120363 0.086262
3868 0.000000 0.249919 0.145437 ... 0.000000 0.000000 0.000000
3276 0.171093 0.108869 0.126710 ... 0.000000 0.052617 0.037709
6557 0.077225 0.081898 0.000000 ... 0.000000 0.000000 0.085103
... ... ... ... ... ... ... ...
14220 0.095894 0.081358 0.023673 ... 0.031184 0.000000 0.105677
4714 0.051262 0.244638 0.031636 ... 0.041675 0.000000 0.000000
Type here to search
1:58 PM
6/3/2021

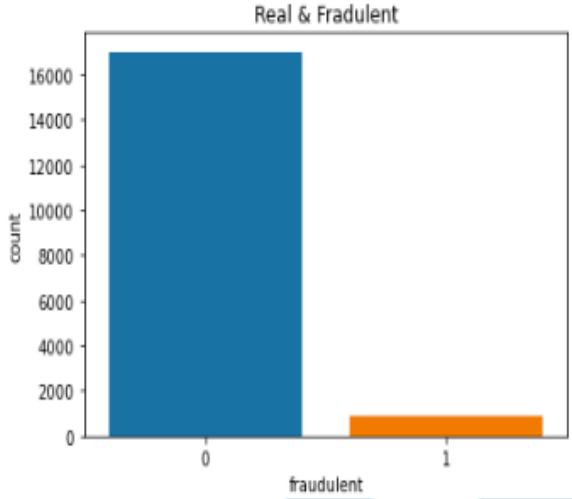

New Tab
Home Page - Select or creat
newproject fakejobe detecti
fakenews on social media -
newproject fakejobe detecti
localhost:8888/notebooks/newproject%20fakejobe%20detection.ipynb
jupyter newproject fakejobe detection Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
14220 0.095894 0.081358 0.023673 ... 0.031184 0.000000 0.105677
4714 0.051262 0.244638 0.031636 ... 0.041675 0.000000 0.000000
10391 0.243186 0.051581 0.060034 ... 0.000000 0.074788 0.000000
2033 0.129287 0.027422 0.000000 ... 0.000000 0.039760 0.028495
12007 0.085699 0.000000 0.079335 ... 0.000000 0.065888 0.094442
with work working world years you your
7081 0.237721 0.051727 0.000000 0.029332 0.066057 0.055841 0.000000
3448 0.163644 0.080712 0.053851 0.000000 0.051536 0.000000 0.000000
3868 0.197064 0.242990 0.000000 0.000000 0.310304 0.000000 0.000000
3276 0.228918 0.000000 0.047082 0.000000 0.090116 0.266627 0.135138
6557 0.193733 0.079628 0.000000 0.000000 0.000000 0.085961 0.101660
... ... ... ... ... ... ... ...
14220 0.128303 0.158205 0.026389 0.000000 0.050508 0.234832 0.126236
4714 0.085733 0.079285 0.000000 0.000000 0.000000 0.000000 0.000000
10391 0.162689 0.100302 0.066921 0.000000 0.000000 0.108279 0.064027
2033 0.194606 0.026662 0.000000 0.000000 0.034048 0.086348 0.068079
12007 0.017916 0.000000 0.000000 0.000000 0.028211 0.166940 0.169224
[12516 rows x 100 columns]
In [92]:
#compare the actualvalue and pred
pred = rfc.predict(X_test)
score=accuracy_score(Y_test,pred)
score
Out[92]: 0.9690529455630127
Type here to search
1:58 PM
6/3/2021


New Tab
Home Page - Select or creat
newproject fakejobe detecti
fakenews on social media -
newproject fakejobe detecti
localhost:8888/notebooks/newproject%20fakejobe%20detection.ipynb
jupyter newproject fakejobe detection Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
[[5006 88]
[ 259 11]]
In [94]:
#************modelfitting in SVM kernel used rbf
mysvc = svm.SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0, decision_function_shape='ovr', degree=3,
gamma='scale', kernel='rbf', max_iter=-1, probability=False,
random_state=None, shrinking=True, tol=0.001,verbose=False).fit(X_train, y_train)
In [96]:
mypredictionsvc=mysvc.predict(X_test)
In [97]:
score1=accuracy_score(y_test,mypredictionsvc)
print(score1)
0.9496644295302014
In [98]:
print(classification_report(y_test,mypredictionsvc))
precision recall f1-score support
0 0.95 1.00 0.97 5094
1 0.00 0.00 0.00 270
accuracy 0.95 5364
macro avg 0.47 0.50 0.49 5364
weighted avg 0.90 0.95 0.93 5364
Type here to search
2:00 PM
6/3/2021

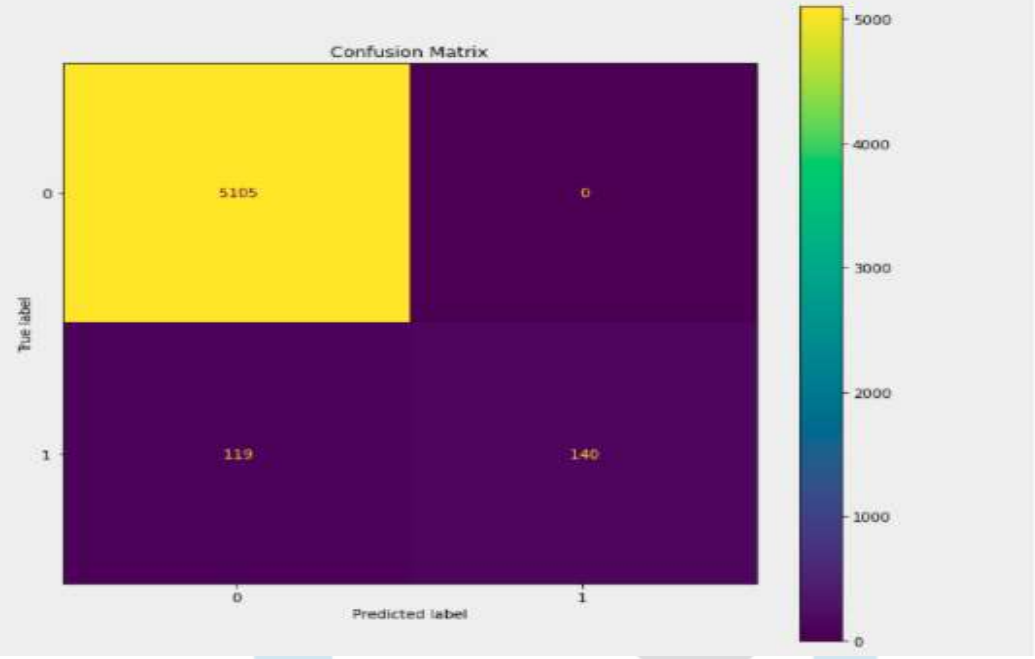
```
#Fraud and Real visualization
sns.countplot(data.fraudulent).set_title('Real & Fradulent')
data.groupby('fraudulent').count()['title'].reset_index().sort_values(by='title',ascending=False)
```

```
/Users/Krishna/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:36: FutureWarning:
ing variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `da
g other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

| | fraudulent | title |
|---|---|---|
| 0 | 0 | 17014 |
| 1 | 1 | 866 |



```
fig, ax = plt.subplots(figsize=(10, 10))
plot_confusion_matrix(pipe, X_test, y_test, values_format='
plt.title('Confusion Matrix')
plt.show()
```




## 6. TESTING

### 6.1 Types of Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is also done to find the errors, gaps and missing functionalities or requirements for a given system.

For our project, we have implemented Integration testing and Unit testing methodologies to test its working. In the case of unit testing, we tested every individual module in order to verify if the individual parts are correct. We have tested the individual internal code modules. In the case of Integration testing, we tested the code for each module and then integrated them based on the hierarchy to top and finally tested the entire code to verify whether it passed all of the test cases. While performing Integration testing, we identified few bugs in final code and then we modified the code and retested it again and again until it satisfied all test cases.

### 6.1 Types of Testing

Testing is broadly classified in to two different types:

- Functional Testing
- Non Functional Testing

#### 6.1.1. Functional Testing:

It is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Some of the Functional Testing are as follows

- Unit Testing
- Smoke Testing
- Sanity Testing
- Integration Testing
- White box testing
- Black Box testing
- User Acceptance testing
- Regression Testing

### 6.1.2. Non Functional Testing:

It is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per non functional parameters which are never addressed by functional testing.

Some of the Non Functional Testing are as follows

- Performance Testing
- Load Testing
- Failover Testing
- Compatibility Testing
- Usability Testing
- Stress Testing
- Maintainability Testing
- Scalability Testing
- Volume Testing
- Security Testing
- Disaster Recovery Testing
- Compliance Testing
- Portability Testing
- Efficiency Testing
- Reliability Testing
- Baseline Testing
- Endurance Testing
- Documentation Testing
- Recovery Testing
- Internationalization Testing
- Localization Testing

**6.2 List of Test Cases**

| S.NO | Test Name | Input | Expected output | Actual Output | Test Case Result |
|---|---|---|---|---|---|
| 1 | Loading the dataset | Dataset | Dataset Loaded | Dataset read | PASS |
| 2 | Split dataset | Training and testing dataset | Splitting of the data into training and testing | Dividing data into training and testing | PASS |
| 3 | Training the model | Training set, parameters | Trained with the provided set | Trained model | PASS |
| 4 | Validation of the model | No of entries from testing data | Validation of the model with best fit | Model generated | PASS |
| 5 | Predicting the accuracy | Accuracy metrics | Predicted accuracy | Accuracy Predicted | PASS |
| 6 | Test Data | Test Column in data | Prediction for the given test case | Predicted result | PASS |
| **7** | Test Case with invalid input | Invalid input values | Enter valid data | Please enter valid data | PASS |

| | | | | | |
|---|---|---|---|---|---|
| 8 | Starting up with the prediction of fake job recruitment | Prediction of job recruitment | Redirecting to the analysis of fake job recruitment | Redirected to analysis of fake job recruitment | PASS |
| 9 | Prediction of the job recruitment analysis | Input values | Resulting Prediction | Prediction Provided | PASS |

Table 1. Test cases of Application

## 7. CONCLUSION

### 7.1 Conclusion

By this project we analysed fake job posting dataset and used machine learning techniques to classify job advertisements which are fraudulent or real. Various machine learning algorithms and performance metrics are used in this project. The benefaction of this Project is one encompassing of contextual capabilities space, which relieved absorbing improvements of accuracy, precision and recall.

By this project we can see that the Support Vector Classifier has given the accuracy score of 97.78% and for 5354 test observations, it has correctly predicted the class labels for 5245 job postings.

### 7.2 Future Scope

In today's context the majority of the companies are recruiting from online platforms. Therefore, many of the applicants are suffering from the fake job posts resulting in losing their data and money. The applicants are facing difficulties in identifying the fake job posts. so, in order to identify the fake job posts we had come up with a machine learning model which analyses the description of the job posts and predicts weather the job post is fake or not.

## 8. BIBLIOGRAPHY


1. B. Alghamdi and F. Alharby, ―An Intelligent Model for Online Recruitment Fraud Detection," J. Inf. Secur., vol. 10, no. 03, pp. 155–176, 2019, doi: 10.4236/jis.2019.103009.
2. D. E. Walters, ―Bayes's Theorem and the Analysis of Binomial Random Variables,‖ Biometrical J., vol. 30, no. 7, pp. 817–825, 1988, doi: 10.1002/bimj.4710300710
3. F. Murtagh, ―Multilayer perceptrons for classification and regression,‖ Neurocomputing, vol. 2, no. 5–6, pp. 183–197, 1991, doi: 10.1016/0925-2312(91)90023-5.
4. P. Cunningham and S. J. Delany, ―K -Nearest Neighbour Classifiers,‖ Mult. Classif. Syst., no. May, pp. 1–17, 2007, doi: 10.1016/S0031-3203(00)00099-6. '
5. H. Sharma and S. Kumar, ―A Survey on Decision Tree Algorithms of Classification in Data Mining,‖ Int. J. Sci. Res., vol. 5, no. 4, pp. 2094–2097, 2016, doi: 10.21275/v5i4.nov162954.
6. E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems,‖ Heliyon, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.
7. L. Breiman, ―ST4_Method_Random_Forest,‖ Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1017/CBO9781107415324.004.
8. B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, ―Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6713 LNCS, pp. 350–359, 2011, doi: 10.1007/978-3-642-21557-5_37.

9 . Natekin and A. Knoll, ―Gradient boosting machines, a tutorial,‖ Front. Neurorobot., vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.

[10] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, "Detection of harassment on web 2.0," Proceedings of the Content Analysis in the WEB, vol. 2, pp. 1–7, 2009.
[11] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying." in The Social Mobile Web, 2011.
[12] V. Nahar, X. Li, and C. Pang, "An effective approach for cyberbullying detection," Communications in Information Science and Management Engineering, 2012.
[13] M. Dadvar, F. de Jong, R. Ordelman, and R. Trieschnigg, "Improved cyberbullying detection using gender information," in Proceedings of the 12th -Dutch-Belgian Information Retrieval Workshop (DIR2012). Ghent, Belgium: ACM, 2012.
[14] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in Advances in Information Retrieval. Springer, 2013, pp. 693–696.
[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," The Journal of Machine Learning Research, vol. 11, pp. 3371–3408, 2010.
16] Q. Huang, V. K. Singh, and P. K. Atrey, "Cyber bullying detection using social and textual analysis," in Proceedings of the 3rd International Workshop on Socially-Aware Multimedia. ACM, 2014, pp. 3–6.

## APPENDIX-A: Python Modules

**About Python:**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Python is simple to use, but it is a real programming language, offering much more structure and support for large programs than shell scripts or batch files can offer. On the other hand, Python also offers much more error checking than C, and, being a very-high-level language, it has high-level data types built in, such as flexible arrays and dictionaries. Because of its more general data types Python is applicable to a much larger problem domain than Awk or even Perl, yet many things are at least as easy in Python as in those languages. Python allows you to split your program into modules that can be reused in other Python programs. It comes with a large collection of standard modules that you can use as the basis of your programs — or as examples to start learning to program in Python. Some of these modules provide things like file I/O, system calls, sockets, and even interfaces to graphical user interface toolkits like Tk. Python is an interpreted language, which can save you considerable time during program development because no compilation and linking is necessary. The interpreter can be used interactively, which makes it easy to experiment with features of the language, to write throw-away programs, or to test functions during bottom-up program development. It is also a handy desk calculator. Python enables programs to be written compactly and readably. Programs written in Python are typically much shorter than equivalent C, C++, or Java programs, for several reasons:

• the high-level data types allow you to express complex operations in a single statement;
• statement grouping is done by indentation instead of beginning and ending brackets;
• no variable or argument declarations are necessary.

Python is extensible: if you know how to program in C it is easy to add a new built-in function or module to the interpreter, either to perform critical operations at maximum speed, or to link Python programs to libraries that may only be available in binary form (such as a vendor-specific graphics library). Once you are really hooked, you can link the Python interpreter into an application written in C and use it as an extension or command language for that application.

**About NumPy:**

NumPy is mostly written in C language, and it is an extension module of Python. It is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single-dimensional array elements. The calculations using NumPy arrays are faster than the normal Python array.

It is also capable of handling a vast amount of data and convenient with Matrix multiplication and data reshaping.

**About Pandas:**

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. It is built on top of the NumPy package, which means **NumPy** is required for operating the Pandas. The name of Pandas is derived from the word **Panel**

**Data**, which means **an Econometrics from Multidimensional data**. It is used for data analysis in Python and developed by **Wes McKinney in 2008**.

Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., **load, manipulate, prepare, model, and analyze**.

**About Sklearn:**
Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. Please note that sklearn is used to build machine learning models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.) .Scikit-learn comes loaded with a lot of features. Supervised learning algorithms: Think of any supervised machine learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikit-learn. I started using scikit to solve supervised learning problems and would recommend that to people new to scikit / machine learning as well. Cross-validation: There are various methods to check the accuracy of supervised models on unseen data using sklearn. Unsupervised learning algorithms: Again there is a large spread of machine learning algorithms in the offering – starting from clustering, factor analysis, principal component analysis to unsupervised neural networks. Various toy datasets: This came in handy while learning scikit-learn. I had learned SAS using various academic datasets (e.g. IRIS dataset, Boston House prices dataset). Having them handy while learning a new library helped a lot. Feature extraction: Scikit-learn for extracting features from images and text (e.g. Bag of words).

## APPENDIX-B:
## UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling

language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

**UML Concepts**

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software intensive system.

The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

**Building Blocks of the UML:**

The vocabulary of the UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.

**Things in the UML:**

There are four kinds of things in the UML:

- Structural things
- Behavioral things
- Grouping things
- Annotational things

Structural things are the nouns of UML models. The structural things used in the project design are:
First, a **class** is a description of a set of objects that share the same attributes, operations, relationships and semantics.
Second, a **use case** is a description of set of sequence of actions that a system performs that yields an observable result of value to particular actor.

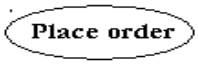



**Fig: Use Cases**

Third, a node is a physical element that exists at runtime and represents a computational resource, generally having at least some memory and often processing capability.

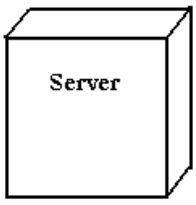
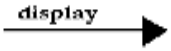



**Fig: Nodes**

**Behavioral things** are the dynamic parts of UML models. The behavioral thing used is:
**Interaction:**
An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. An interaction involves a number of other elements, including messages, action sequences (the behavior invoked by a message, and links (the connection between objects).




**6.1.3 Relationships in the UML:**

There are four kinds of relationships in the UML:

- Dependency
- Association

- Generalization
- Realization

A **dependency** is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing (the dependent thing).

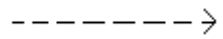


**Fig: Dependencies**

An **association** is a structural relationship that describes a set links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts.



**Fig: Association**

A **generalization** is a specialization/ generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element(the parent).

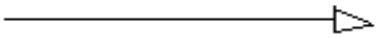


**Fig: Generalization**

A **realization** is a semantic relationship between classifiers, where in one classifier specifies a contract that another classifier guarantees to carry out.

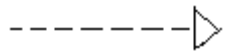


**Fig: Realization**

**Sequence Diagrams:**

UML sequence diagrams are used to represent the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram.

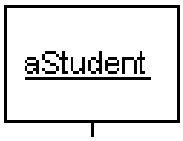
Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

**Actor**

Represents an external person or entity that interacts with the system

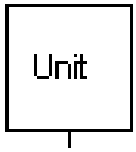

**Object**

Represents an object in the system or one of its components




**Unit**

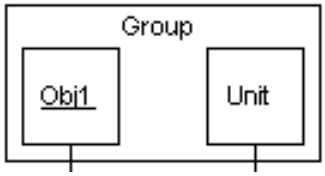
Represents a subsystem, component, unit, or other logical entity in the system (may or may not be implemented by objects)



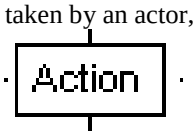

**Separator**

Represents an interface or boundary between subsystems, components or units (e.g., air interface, Internet, network)

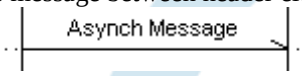
**Group**
Groups related header elements into subsystems or components



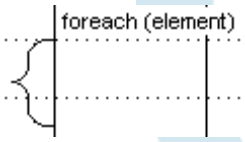

Sequence Diagram Body Elements
**Action**
Represents an action taken by an actor, object or unit




**Asynchronous Message**
An asynchronous message between header elements



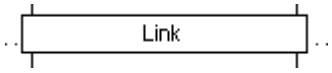

**Block**
A block representing a loop or conditional for a particular header element



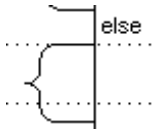

**Call Message**
A call (procedure) message between header elements


**Create Message**
A "create" message that creates a header element (represented by lifeline going from dashed to solid pattern)

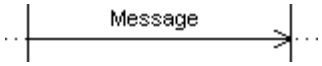

**Diagram Link**
Represents a portion of a diagram being treated as a functional block. Similar to a procedure or function call that abstracts functionality or details not shown at this level. Can optionally be linked to another diagram for elaboration.




Else Block Represents an "else" block portion of a diagram block




**Message**
A simple message between header element




**Return Message**
A return message between header elements

**PLAGIARISM REPORT**

| | |
|---|---|
| **Report Title:** | FAKE JOB RECRUITMENTS DETECTION |
| **Report Link:** (Use this link to send report to anyone) | https://www.check-plagiarism.com/plag-report/2360414aca6a7baf3b92760dbeac97aa5f97f161577975454 |
| **Report Generated Date:** | 10-06-2022 |
| **Total Words:** | 3490 |
| **Total Characters:** | 23171 |
| **Keywords/Total Words Ratio:** | 81.9% |
| **Excluded URL:** | No |
| **Unique:** | 91% |
| **Matched:** | 9% |