A Project Report

on

Fake Job Posting Detection Using Machine Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

S.Sameena nazmi	(174G1A0577)
B.Sai Pranathi	(174G1A0570)
K.Souri Vandana	(174G1A0585)
J.Sai Teja	(174G1A05B8)

Under the Guidance of

Mr. Lingam Suman M.Tech., (Ph.D)

Assistant Professor

SRIT
Empowering Knowledge

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY :: ANANTHAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with ‘A’ Grade, Approved by AICTE, New Delhi & Accredited by NBA(EEE,ECE&CSE))

2017-2021

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA(EEE,ECE&CSE))
Rotarypuram Village , B K Samudram Mandal , Ananthapuramu – 515701

Certificate

This is to certify that the project report entitled Fake Job Posting detection Using Machine Learning is the bonafide work carried out by **S.Sameena nazmi** bearing Roll Number 174G1A0577**,**B.Sai Pranathi bearing Roll Number 174G1A0570**,** K.Souri Vandana bearing Roll Number 174G1A0585 and J.Sai Teja bearing Roll Number 174G1A05B8 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2020-2021.

Guide

Mr. Lingam Suman M.Tech., (Ph.D)
Assistant Professor

Head of the Department

Dr. G.K.V. Narasimha Reddy Ph.D
Professor & HOD

Date:

EXTERNAL EXAMINER

Ananthapuramu

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mr. Lingam Suman,M.Tech(Ph.D) Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Dr.P.Chitralingappa,M.Tech,Ph.D, Assistant Professor & Mrs.M.Sowmya,M.Tech, Assistant Professor ,** project coordinators valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Dr.G.K.V.Narasimha Reddy,Ph.D, Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr.G.BalaKrishna,M.Tech,Ph.D, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

Declaration

We, Ms S.Sameena Nazmi with reg no: 174G1A0577, Ms B.Sai Pranathi with reg no: 174G1A0570, Ms K.Souri Vandana with reg no: 174G1A0585, Mr J.Sai Teja with reg no: 174G1A0584 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled“FAKE JOB POSTING DETECTION USING MACHINE LEARNING” embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Mrs T.Divya Vani M.Tech, Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfilment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

S.SAMEENA NAZMI	Reg no: 174G1A0577
B.SAI PRANATHI	Reg no: 174G1A0570
K.SOURI VANDANA	Reg no: 174G1A0585
J.SAI TEJA	Reg no: 174G1A05B8

Contents

List of Figures

List of Screeens

List of Abbreviations

CSV	Comma-separated Values
SFS	Sequential Feature Selection
SRS	Software Requirement Specification
UML	United Modelling Language
Numpy	Numerical Python
ML	Machine Learning

ABSTRACT

To avoid fraudulent post for job in the internet, an application using machine learning based classification techniques is proposed in the project. 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.

CHAPTER 1

INTRODUCTION

1.1 Fake Job

These days recruitments are mainly done online through online portals such as naukri.com,,monster.com. Organizations put their job advertisement with desired skills required on these portals. Job seekers or candidates put their resumes and skill details on these portals. Now, companies can scan the profiles of desired candidates and contact the candidates as well as candidates can also apply to the job profiles in which they are interested. After first screening, companies contact the shortlisted candidates for further processing and recruit the suitable candidates. Online recruitment is beneficial for both candidates as well as the companies. In Dec 2016, Naukri.com had a database of about 49.5 million registered users, 11000 resumes were getting added daily. This shows the impact that these online job portals have on users. The online recruitment is beneficial for both recruiter as well as candidates. However, in the recent years scammers have started this online recruitment industry which has given a new type of fraud, i.e., Online Recruitment Fraud (ORF). In ORF spammers give lucrative job offers to the candidates and steal their money and private information. ORF not only harms the users but it is also problematic for the companies. As, it damages the reputation of companies and leaves a negative impact in the mind of job seekers about the given company.Detection of fraud job offers from a legitimate set of job is a technically challenging problem. The main challenge the class imbalance problem as the number of fraud jobs are relatively less as compared to the legitimate jobs. This makes learning the features of fraud jobs for automated prediction a challenging task.

1.2 Problem Definition

There are a lot of job advertisements on the internet, even on the reputed job advertising sites, which never seem fake. But after the selection, the so-called recruiters start asking for the money and the bank details**.**

- Many of the candidates fall in their trap and lose a lot of money and the current job sometimes. So, it is better to identify whether a job advertisement posted on the site is real or fake**.**
- Identifying it manually is very difficult and almost impossible.

1.3 Project Purpose

- To avoid fraudulent post for job in the internet, an application using machine learning based classification techniques is proposed in the project.
- Random Forest classifiers are used for training fraudulent post in the dataset and the results of those classifiers are used for identifying the best employment scam detection model.
- It helps in detecting fake job posts from an enormous number of posts.
- After model is trained test file is given as input with features and verified with model and fake or non fake information is stored to csv file.

1.4 Project Features

- The features of Fake job prediction Machine Learning are as follows.
- machine learning approach is applied which employs several classification algorithms for recognizing fake posts. In this case, a classification tool.

1.5 Machine Learning

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by,improves with experience”.Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.Machine learning tasks Machine learning tasks are typically classified into several broad categories:

Supervised learning: The computer is presented with example inputs and theirdesired outputs, given by a "teacher", and the goal is to learn a general rule thatmaps inputs to outputs. As special cases, the input signal can be only partiallyavailable, or restricted to special feedback.

Active learning: The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle orplaying a game against an opponent.

1.6 Features of Machine Learning

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data withouthelp of humans or any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.

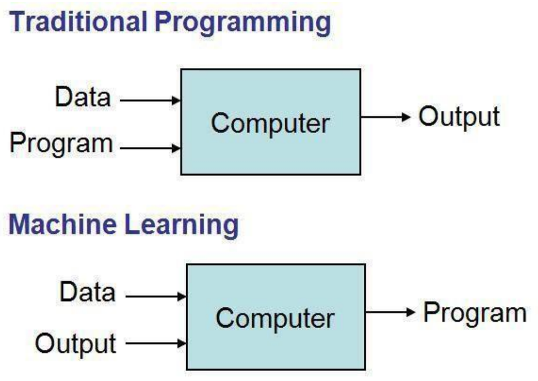


Fig.1.6.1. Traditional Programming vs Machine Learning

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which wewill provide us the exact solution in predicting the Fake Job Prediction of the Users.

How Does Machine Learning Works?

- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.

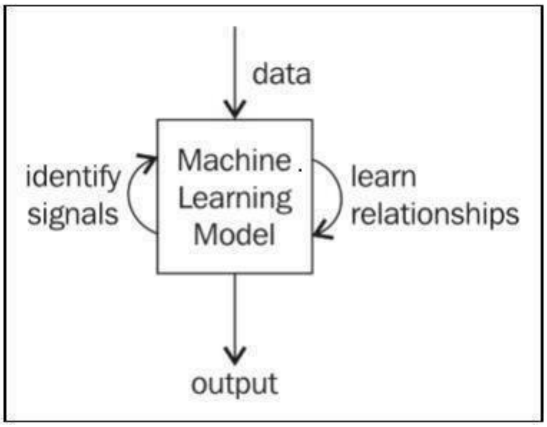


Fig.1.6.2. Overview of Machine Learning model

- There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration,robotics, social networks, debugging and much more.
- There are 3 types of machine learning supervised, unsupervised, and reinforcement.

1.7 Random Forest Classifier

Random Forest Classifier: Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output.

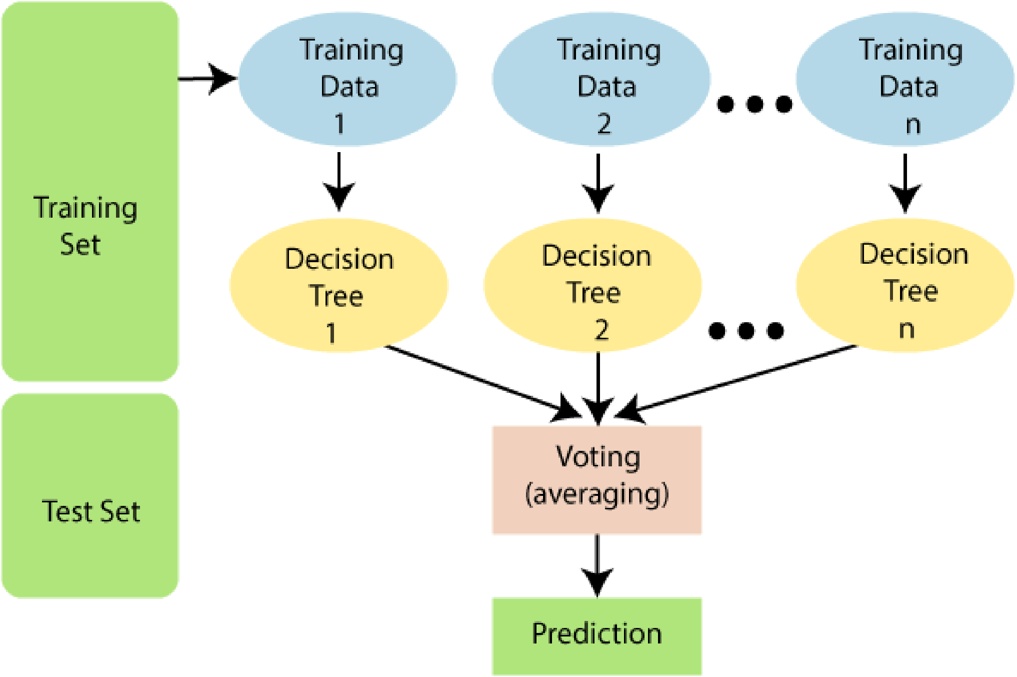


Fig.1.6.3. Trees in Random Forest

The greater number of trees in the forest leads to higher accuracy and prevents theproblem of overfitting. Two types of randomness are built into the trees.

- Each tree is built on a random sample from the original data.
- At each tree node, a subset of features are randomly selected to generate the best split.

Advantages

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

CHAPTER 2

LITERATURE SURVEY

2.1 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.

2.2 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 prediction may be broadly categorized into - Single Classifier based Prediction and Ensemble Classifiers based Prediction.

2.3 Software Description

PYTHON

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle- detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's developers strive to avoid premature optimization, and reject patches to non- critical parts of CPython that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functionsto extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes directC-level API calls into the Python interpreter.An important goal of Python's developers is keeping it fun to use. Python's design offers some support for functional programming inthe Lisp tradition. It has filter, map, and reduce functions, list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Benefits of Python

- Presence of Third-Party Modules.
- Extensive Support Libraries.
- Open Source and Community Development.
- Learning Ease and Support Available.
- User-friendly Data Structures.
- Productivity and Speed.
- Highly Extensible and Easily Readable Language.

.

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

- Elicitation – I ask questions, you talk, I listen
- Validation – I analyze, I ask follow-up questions
- Specification – I document, I ask follow-up questions
- Verification – We all agree Most of the work in the Analysis Phase isperformed by the role of analyst.

3.2 Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.Breast Cancer Detection Using Machine Learning Computer Science and Engineering, SRIT Page 16 of 46 The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

3.3 Funtional Requirements

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all cases where the system uses the functional requirements arecaptured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability.

- Functional Requirements concerns with the specific functions delivered bythe system.So, Functional requirements are statements of the services that the system must provide.
- The functional requirements of the system should be both complete and consistent
- Completeness means that all the services required by the user should be defined.
- Consistency means that requirements should not have any contradictory definitions.
- The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in details, its inputs and outputs, exceptions and so on.
- Take user id and password match it with corresponding file entries. If a match is found then continue else raise an error message.

3.4 Non-Functional Requirements

- Non-functional Requirements refer to the constraints or restrictions on the system. Theymay relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
- The non-functional requirements can be built on the basis of needs of the user, budgetconstraints, organization policies and etc.
- **Performance requirement:** All data entered shall be up to mark and no flawsshall be there for the performance to be 100%.
- **Platform constraints:** The main target is to generate an intelligent system to predict the height.

- **Accuracy and Precision**: Requirements are accuracy and precision of the data
- **Modifiability:** Requirements about the effort required to make changes in thesoftware. Often, the measurement is personnel effort (person- months).
- **Portability:** Since mobile phone is handy so it is portable and can be carried and used whenever required.
- **Reliability**: Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction, and a strategy for correction.
- **Security**: One or more requirements about protection of your system and its data.
- **Usability:** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

Accessibility

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project peoplewho have registered with the cloudcan access the cloud to store and retrieve their data with the help of a secret key sent to their email ids. User interface is simple and efficient and easy to use.

Mainatainability

In software engineering, maintainability is the ease with whicha software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

Scalability

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

Portability

Portability is one of the key concepts of high-level programming.

Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

Validation

It is the process of checking that a software system meets specifications and that it fulfilsits intended purpose. It may also be referred to as software quality comtrol.It is normallythe responsibility of software testers as part of the software development lifecycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specificationartefacts or as needs of those who will operate the software only; but, as the needs of allthe stakeholders.

3.3 Hardware Requirements

- RAM : 512Mb or above
- Input Device : Keyboard and Mouse
- Output Device : Monitor or PC

Software Requirements

- Operating System : Windows 7, 10
- Coding Language : PYTHON
- Version : PYTHON 3.7

CHAPTER 4

DESIGN

4.1 Design Goals

The Design goals consist of various design which we have implemented inour system Fake Job Prediction using machine learning. This system has built with various designs suchas data flow diagram, sequence diagram, class diagram, use case diagram, component diagram, activity diagram, state chart diagram, deployment diagram. After doing these various diagrams and based on these diagrams we have done our project.

a.Select data from Database

b.Fake job prediction.

4.2 System Architecture

Fake Job Prediction using machine learning predicts the presence of the Fake Job Prediction for the user based on various features and the information the user gives test dataset. The architecture of the system Fake Job Prediction using machine learning consist of various datasets through which we will compare the fake job from test dataset and predicts it, then the datasets are transformed into the smaller sets and from there it gets classified based on the classification algorithms later on the classified data is then processed into the machine learning technologies through which the data gets processed and goes in to the Fake Job Prediction model using all the inputs from the user that is mentioned above**.** Then after user entering the above information and overall processed data combines and compares in the prediction model of the system and finally predicts the Fake Job Prediction. An architecture diagram is a

Fake Job Prediction using machine learning predicts the presence of the Fake Job Prediction for the user based on various features and the information the user gives test dataset. The architecture of the system Fake Job Prediction using machine learning consist of various datasets through which we will compare the fake job from test dataset and predicts it, then the datasets are transformed into the smaller sets and from there it gets classified based on the classification algorithms later on the classified data is then processed into the machine learning technologies through which the data gets processed and goes in to the Fake Job Prediction model using all the inputs from the user that is mentioned above**.** Then after user entering the above information and overall processed data combines and compares in the prediction model of the system and finally predicts the Fake Job Prediction. An architecture diagram is a graphical representation of a set of concepts, that are part of anarchitecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system.

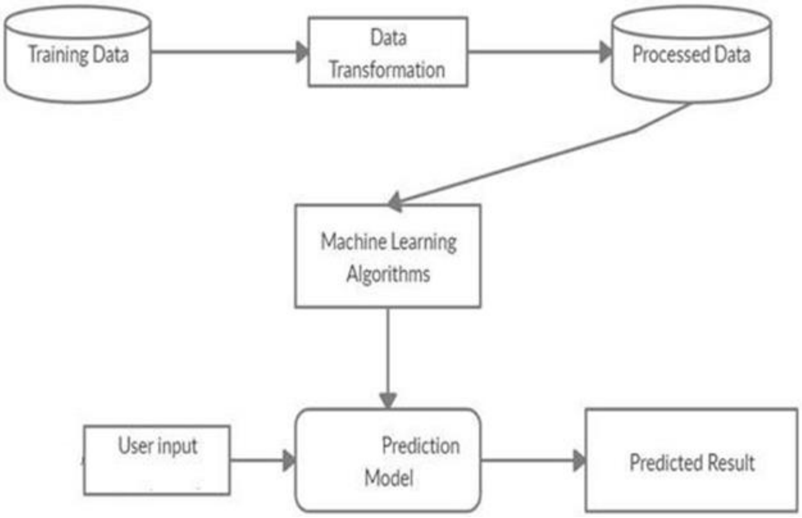


Fig.4.2. System Architecture

4.3 Dataflow Diagram

The dataflow diagram of the project Fake Job Prediction using machine learning consist of all the various aspects a normal flow diagram requires. This dataflow diagram shows how from starting the model flows from one step to another, like how we input dataset process through various steps apply algorithms and test along with the test dataset that goes into the system, compares with the prediction model and if 0 or 1 is predicts theappropriate results .

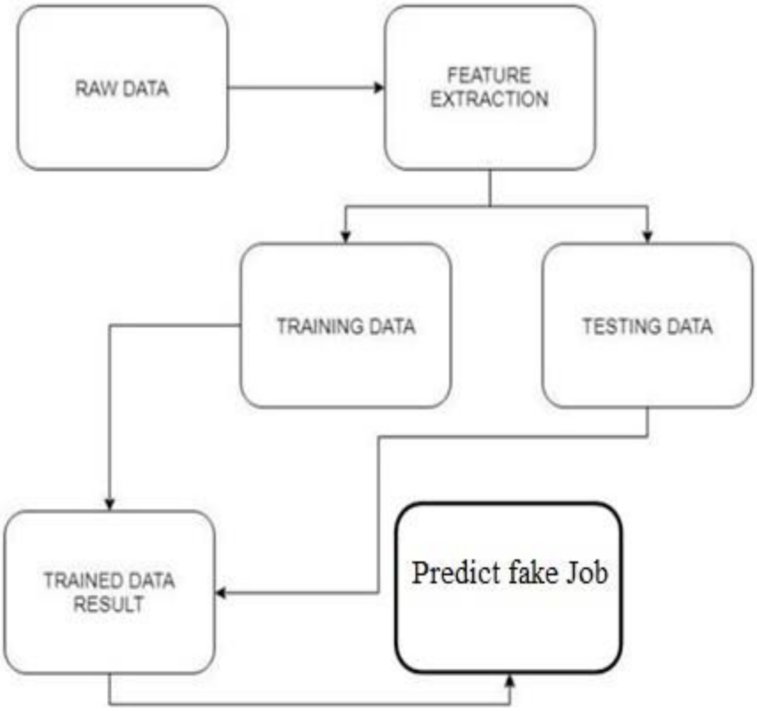


Fig.4.3. Dataflow Diagram

4.4 Use Case Diagram

The Use Case diagram of the project Fake Job Prediction prediction using machine learning consist of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from onestep to another, , like how we input dataset process through various steps apply algorithmsand test along with the test dataset that goes into the system, compares with the prediction model and if 0 or 1 is predicts theappropriate results. Here the use case diagram of all the entities are linked toeach other where the user gets started with the system.

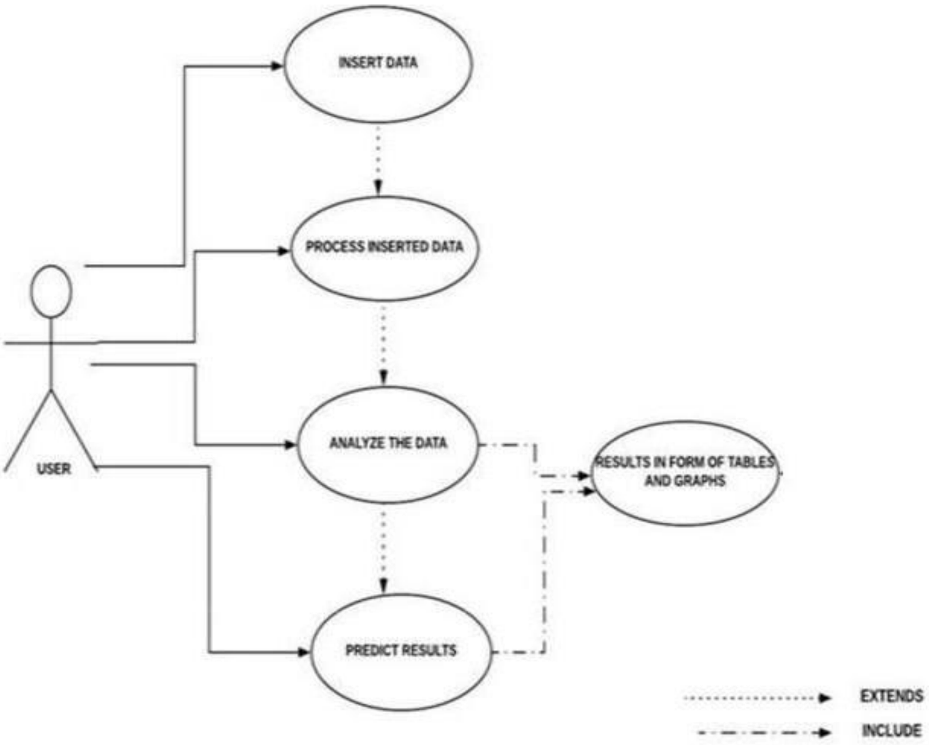


Fig.4.4. Use Case Diagram

4.5Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects ofthe system. Activity diagram is basically a flowchart to represent the flow from one activityto another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. In this diagram two stages of process is explained

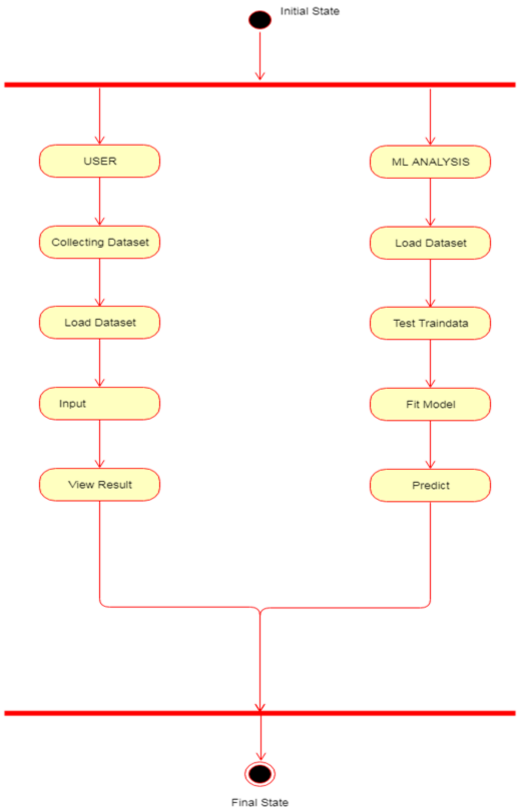


Fig.4.5. Activity Diagram

4.6 Component Diagram

A component diagram,also known as a UML component diagram, describes the organization and wiring of the physical componentsin a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. Here component diagram consists of all major components that is used to built a system. So, Design, Algorithm, File System and Datasets all are linked to one another. Datasets are used to compare the results and algorithm is used to process those results and give a correct accuracy and design UI is used to show the result in an appropriate way in the system and file system is used to store the user data. So, like this all components are interlinked to each other.

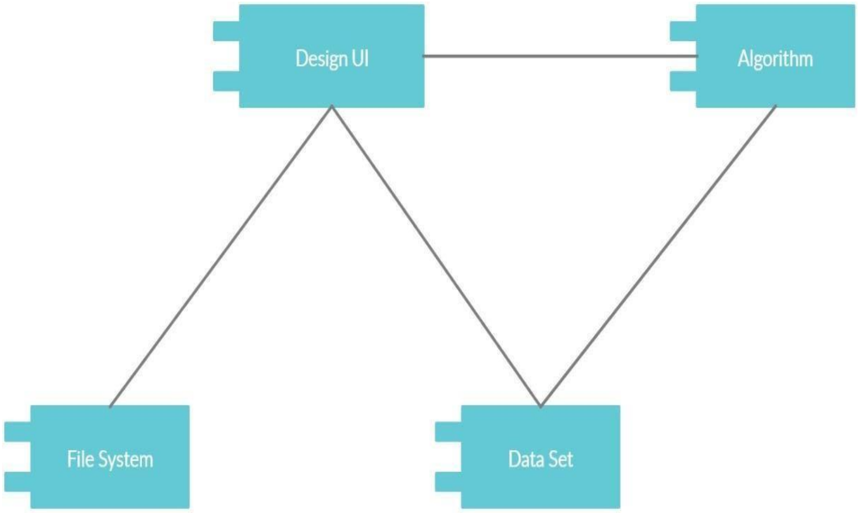


Fig.4.6. Component Diagram

4.7 Deployment Diagram

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. Here the deployment diagram show the final stage of the project and it also shows how the model looks like after doing all the processes and deploying in the machine. Starting from the system how it processes the user entered information and then comparing that information with the help of datasets, then training and testing those data using the algorithms such as decision tree, naïve Bayes, random forest. Then

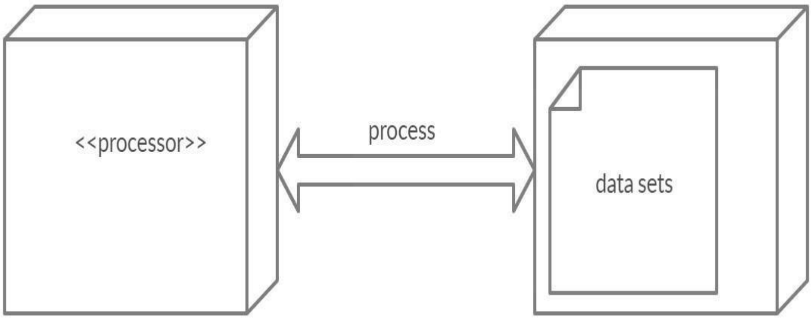


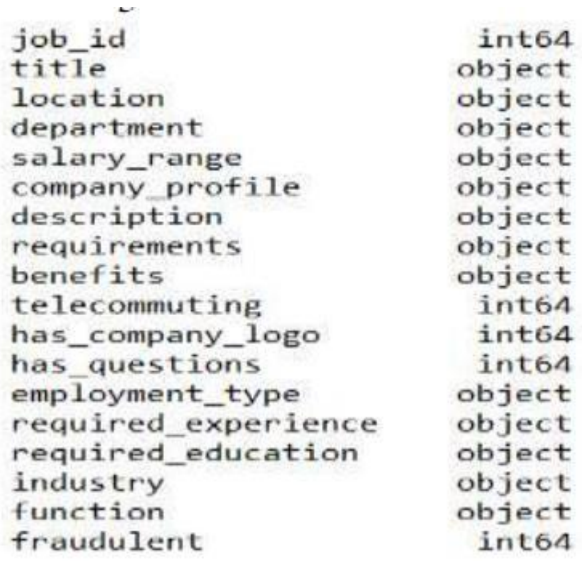
Fig.4.7. Deployment Diagrams

CHAPTER 5

IMPLEMENTATION

5.1 Overview

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 besuspicious. The dataset has the schema as shown in Fig 1.



```
job_id                      int64
title                      object
location                   object
department                 object
salary_range               object
company_profile            object
description                object
requirements               object
benefits                   object
telecommuting               int64
has_company_logo            int64
has_questions               int64
employment_type            object
required_experience        object
required_education         object
industry                   object
function                   object
fraudulent                  int64
```

Fig.5.1. Fake job posting Features

This dataset contains of job posts. 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 removal. This prepares the dataset to be transformed into categorical encoding in order to obtain a feature vector. This feature vectors are fitted to several classifiers. The following diagram Fig. 2 depicts a description of the working paradigm of a classifier for prediction. A ML classifier Random Tree Classifier is applied for classifying job post as fake. It is to be noted that the attribute _fraudulent ‘of the dataset is kept as target class for classification purpose. At first, the classifiers are trained using the 80% of the entire dataset and later 20% of the entire dataset is used for the prediction purpose. The performance measure metrics such as Accuracy.

Implementation of Classifiers

In this framework classifiers are trained using appropriateparameters. For maximizing the performance of these models, default parameters may not be sufficient enough. Adjustment of these parameters enhances the reliability of this model which may be regarded as the optimised one for identifying as well as isolating the fake job posts from the job seekers. This framework utilized RFC classifier. The Random Forest classifier gives a promising result for the. After constructing these classification models, training data are fitted into it. Later the testing dataset are used for prediction purpose. After the prediction is done, performance of the classifiers are evaluated based on the predicted value and the actual value.

5.2 Random Forset Algorithm

- It is an ensemble classifier using many decision trees models; it can be used for regression as well as classification.
- Accuracy and variable importance information can be provided with the results.
- A random forest is the classifier consisting of a collection of tree structured classifiers k, where the k is independently, identically distributed random trees and each random treeconsist of the unit of vote for classification of input.
- Random forest uses the Gini index for the classification and determining the final class in each tree.
- The final class of each tree is aggregated and voted by the weighted values to constructthe final classifier.
- The working of random forest is, A random seed is chosen which pulls out at a random, a collection of samples from the training datasets while maintaining the class distribution.

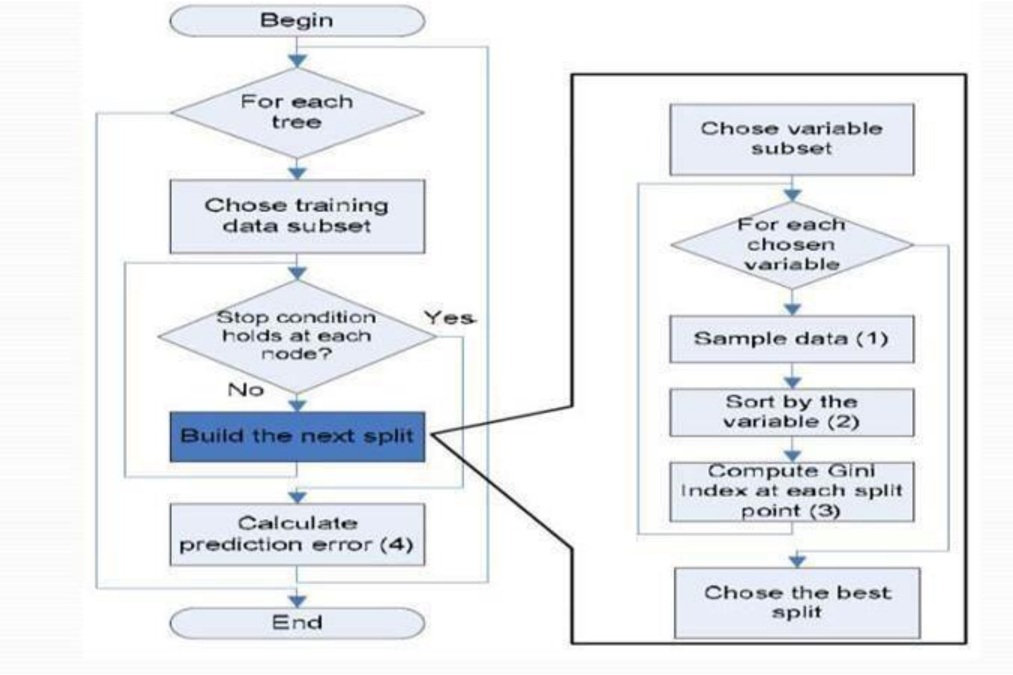


Fig 5.3. Random Forest Example

Modules

Collecting Dataset

In this step data set with are considered as input has many fields in that except last one all are called as features and last cell is called as labels which arein to 0 1 format.

Data set processing:

Given dataset is not in required format so we need to remove unwanted fields from the dataset and use them as features and process data for scalar format.

Preprocessing:

We convert data in to scalar format and then create new features which are passed to algorithm and features are saved in x and labels in y.

Algorithm Fit:

In this step train features and labels are fit to algorithm and model is saved to system which is used for prediction.

Prediction:

In this step details are fed as input in the form of csv of various profiles and prediction is performed.

Steps to implement Random Forest Classifier in Python

Step 1 - Import the Libraries

We will start by importing the necessary libraries required to implement the Algorithm in Python. We will import the numpy libraries for scientific calculation.

Step 2 - Fetch the Data

We will fetch the data from csv file using ‘**pandas_datareader**’. We storethis in a data frame ‘df’

Step 3 - Split the Dataset

We will split the dataset into **training dataset** and **test dataset**. We will use 70% of our data to train and the rest 30% to test. To do this, we will create a split parameter which will divide the dataframe in a 70-30 ratio.

Step 4 - Instantiate RandomForestClassifier Model

After splitting the dataset into training and test dataset, we will instantiate RandomForestClassifier fit the train data by using ‘**fit**’ function. Then we will store as model.

Step 5 – Prediction

New input csv file with different job profile details are given as input and prediction is performed and details are stored in new csv with prediction results.

Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

NumPy

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since, arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit- learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem. NumPy provides the essential multi-dimensional array- oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using import numpy as np.

Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivottables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd.

Pip

The pip command is a tool for installing and managing Python packages, such as those found in the Python Package Index. It's a replacement for easy install. The easiest way to install the nfl* python modules and keep them up-to- date is with a Python-based package manager called Pip.

Sklearn

Skikit-learn is a free software machine library for Python programming language.It features various classification , regression and clustering algorithms including support vector machine, random forest, k-means and gradient boosting. In our project we have used different features.

- from sklearn.ensemble import RandomForestClassifier:
- Used for Random Forest Classifier algorithm.
- from sklearn.model_selection import train_test_split:

  Used for Splitting the dataset into Training and Testing.
- from sklearn.metrics import accuracy_score as acc

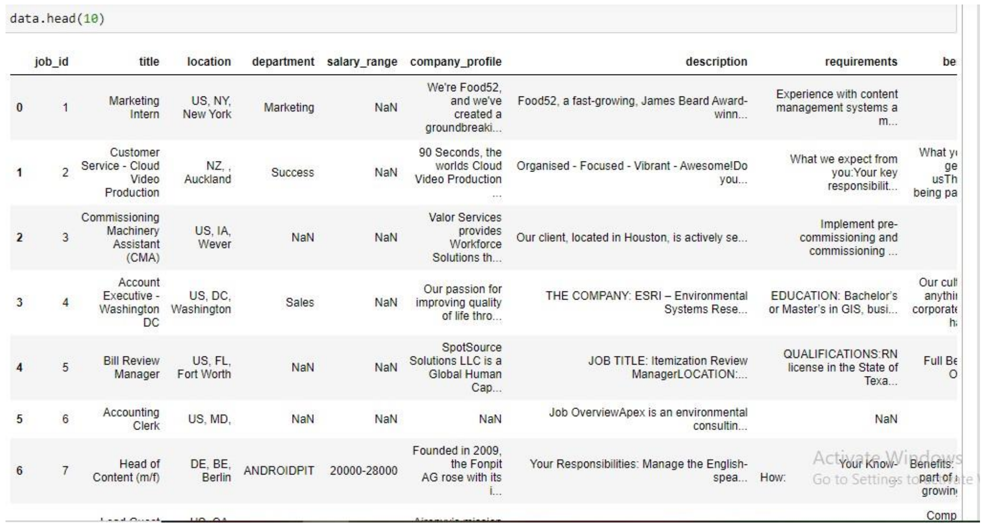
  Used for calculating the Accuracy

Comma-separated values(CSV)

The dataset used in this project is a .CSV file.

In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

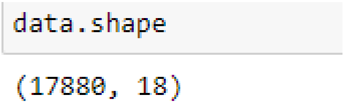
CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma. A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spreadsheets.



```
data.head(10)
```

	job_id	title	location	department	salary_range	company_profile	description	requirements	be
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experience with content management systems a m...	
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What we expect from you:Your key responsibilit...	What y ge usTh being pa
2	3	Commissioning Machinery Assistant (CMA)	US, IA, Wever	NaN	NaN	Valor Services provides Workforce Solutions th...	Our client, located in Houston, is actively se...	Implement pre-commissioning and commissioning ...	
3	4	Account Executive - Washington DC	US, DC, Washington	Sales	NaN	Our passion for improving quality of life thro...	THE COMPANY: ESRI – Environmental Systems Rese...	EDUCATION: Bachelor's or Master's in GIS, busi...	Our cul anythi corporat h
4	5	Bill Review Manager	US, FL, Fort Worth	NaN	NaN	SpotSource Solutions LLC is a Global Human Cap...	JOB TITLE: Itemization Review ManagerLOCATION:...	QUALIFICATIONS:RN license in the State of Texa...	Full Be O
5	6	Accounting Clerk	US, MD,	NaN	NaN	NaN	Job OverviewApex is an environmental consultin...	NaN	
6	7	Head of Content (m/f)	DE, BE, Berlin	ANDROIDPIT	20000-28000	Founded in 2009, the Fonpit AG rose with its i...	Your Responsibilities: Manage the English-spea...	How: Your Know-...	Benefits: part of growin

Fig.5.2.1 Fake Job Posting Dataset



```
data.shape
```

```
(17880, 18)
```

Fig.5.2.2. Fake Job Posting Dataset Shape

Model construction using Random Forest Classifier

Random Forest Classifier: The Random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree. This model uses two key concepts that gives it the name random:

Algorithm

1. Randomly select “K” features from total “m” features where k << m.
2. Among the “K” features, calculate the node “d” using the best split point
3. Split the node into daughter nodes using the best split.
4. Repeat the 1 to 3 steps until number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” number times to to create “n”number of trees.

In the next stage, with the random forest classifier created, we will make the prediction. The random forest prediction pseudocode is shown below:

1. Takes the test features and use the rules of each randomly created decision treeto predict the outcome and stores the predicted outcome (target).
2. Calculate the votes for each predicted target.
3. Consider the high voted predicted target as the final prediction from the randomforest algorithm.

4. model=RandomForestClassifier(n_estimators=100,criterion='entropy', random_state = 1)

Model

Model is a system that answers the question of a problem statement andthis model is created via a process called “training”. The goal of training is to create an accurate model that answers our questions correctly most of the time.

Splitting of data

Splitting of data is dividing your data set into two subsets

- Training set—a subset to train a model which is used for fit and tune themodel
- Testing set—a subset to test the trained model which is used to evaluate themodel of unseen data.

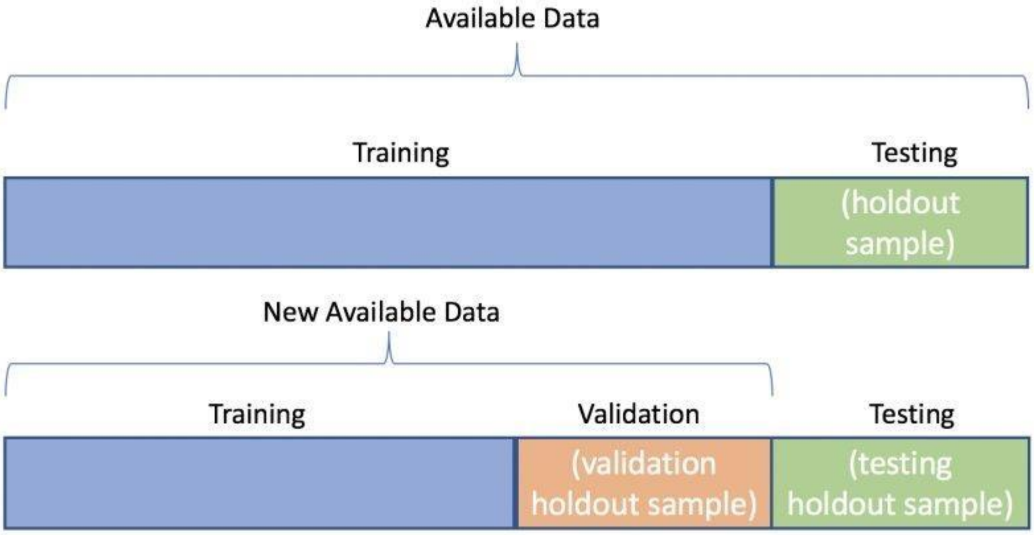
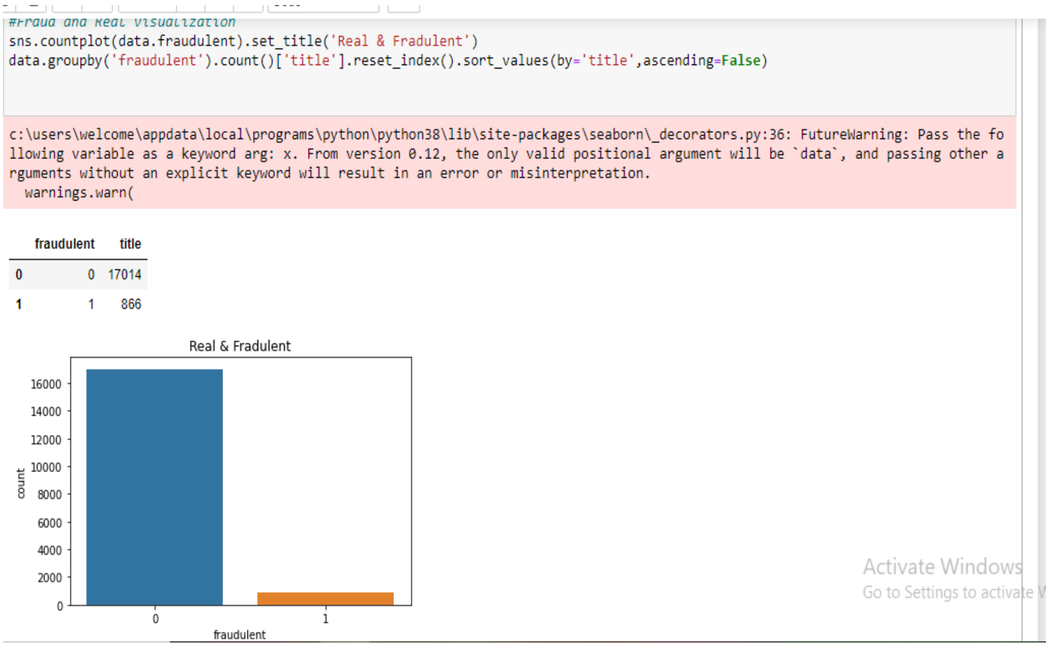


Fig.5.2.3. Dataset Splitting Ratio

Code

Visualizing number of real and fake jobs



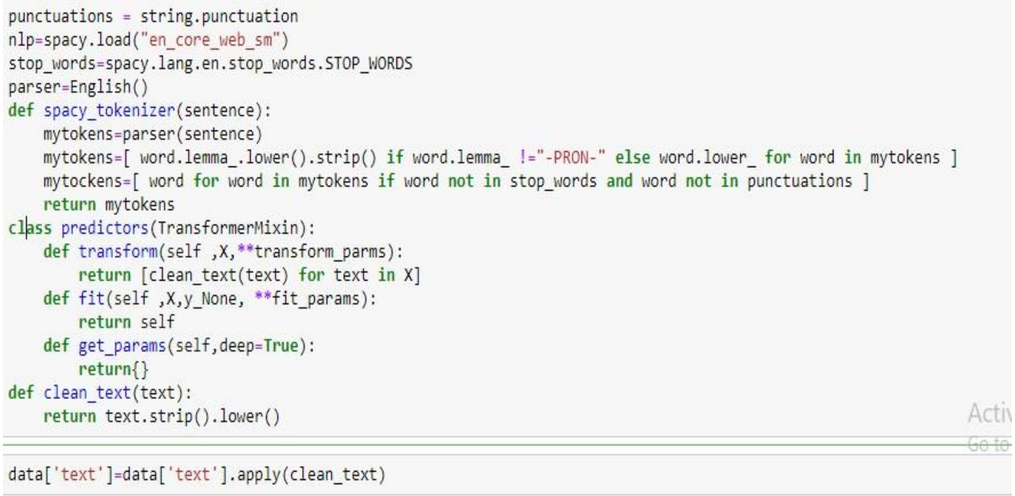
```
#Fraud and Real visualization
sns.countplot(data.fraudulent).set_title('Real & Fradulent')
data.groupby('fraudulent').count()['title'].reset_index().sort_values(by='title',ascending=False)
```

```
c:\users\welcome\appdata\local\programs\python\python38\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the fo
llowing variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

	fraudulent	title
0	0	17014
1	1	866

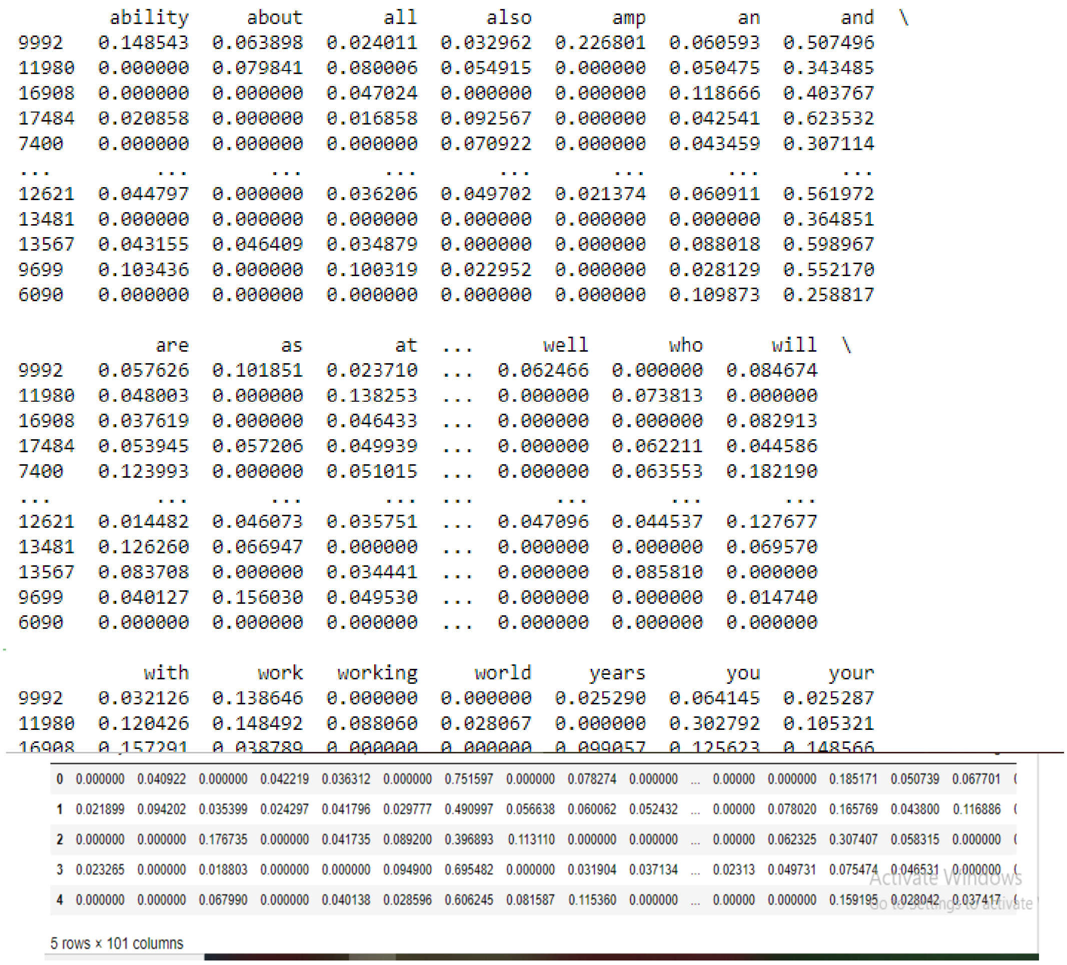
Fig.5.2.4. Visualizing number of real and fake job

Preprocessing the dataset



```
punctuations = string.punctuation
nlp=spacy.load("en_core_web_sm")
stop_words=spacy.lang.en.stop_words.STOP_WORDS
parser=English()
def spacy_tokenizer(sentence):
    mytokens=parser(sentence)
    mytokens=[ word.lemma_.lower().strip() if word.lemma_ !="-PRON-" else word.lower_ for word in mytokens ]
    mytockens=[ word for word in mytokens if word not in stop_words and word not in punctuations ]
    return mytokens
class predictors(TransformerMixin):
    def transform(self ,X,**transform_parms):
        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{}
def clean_text(text):
    return text.strip().lower()
```

```
data['text']=data['text'].apply(clean_text)
```

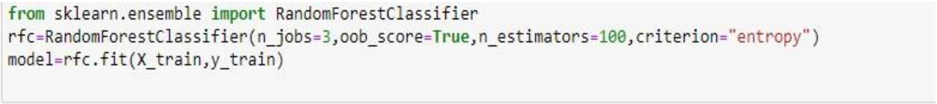


```
         ability     about       all      also       amp        an       and  \
9992    0.148543  0.063898  0.024011  0.032962  0.226801  0.060593  0.507496
11980   0.000000  0.079841  0.080006  0.054915  0.000000  0.050475  0.343485
16908   0.000000  0.000000  0.047024  0.000000  0.000000  0.118666  0.403767
17484   0.020858  0.000000  0.016858  0.092567  0.000000  0.042541  0.623532
7400    0.000000  0.000000  0.000000  0.070922  0.000000  0.043459  0.307114
...          ...       ...       ...       ...       ...       ...       ...
12621   0.044797  0.000000  0.036206  0.049702  0.021374  0.060911  0.561972
13481   0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.364851
13567   0.043155  0.046409  0.034879  0.000000  0.000000  0.088018  0.598967
9699    0.103436  0.000000  0.100319  0.022952  0.000000  0.028129  0.552170
6090    0.000000  0.000000  0.000000  0.000000  0.000000  0.109873  0.258817

             are        as        at  ...      well       who      will  \
9992    0.057626  0.101851  0.023710  ...  0.062466  0.000000  0.084674
11980   0.048003  0.000000  0.138253  ...  0.000000  0.073813  0.000000
16908   0.037619  0.000000  0.046433  ...  0.000000  0.000000  0.082913
17484   0.053945  0.057206  0.049939  ...  0.000000  0.062211  0.044586
7400    0.123993  0.000000  0.051015  ...  0.000000  0.063553  0.182190
...          ...       ...       ...  ...       ...       ...       ...
12621   0.014482  0.046073  0.035751  ...  0.047096  0.044537  0.127677
13481   0.126260  0.066947  0.000000  ...  0.000000  0.000000  0.069570
13567   0.083708  0.000000  0.034441  ...  0.000000  0.085810  0.000000
9699    0.040127  0.156030  0.049530  ...  0.000000  0.000000  0.014740
6090    0.000000  0.000000  0.000000  ...  0.000000  0.000000  0.000000

            with      work   working     world     years       you      your
9992    0.032126  0.138646  0.000000  0.000000  0.025290  0.064145  0.025287
11980   0.120426  0.148492  0.088060  0.028067  0.000000  0.302792  0.105321
16908   0.157291  0.038789  0.000000  0.000000  0.099057  0.125623  0.148566
```

```
0  0.000000  0.040922  0.000000  0.042219  0.036312  0.000000  0.751597  0.000000  0.078274  0.000000  ...  0.00000  0.000000  0.185171  0.050739  0.067701  (
1  0.021899  0.094202  0.035399  0.024297  0.041796  0.029777  0.490997  0.056638  0.060062  0.052432  ...  0.00000  0.078020  0.165769  0.043800  0.116886  (
2  0.000000  0.000000  0.176735  0.000000  0.041735  0.089200  0.396893  0.113110  0.000000  0.000000  ...  0.00000  0.062325  0.307407  0.058315  0.000000  (
3  0.023265  0.000000  0.018803  0.000000  0.000000  0.094900  0.695482  0.000000  0.031904  0.037134  ...  0.02313  0.049731  0.075474  0.046531  0.000000  (
4  0.000000  0.000000  0.067990  0.000000  0.040138  0.028596  0.606245  0.081587  0.115360  0.000000  ...  0.00000  0.000000  0.159195  0.028042  0.037417  (

5 rows × 101 columns
```

Fig.5.2.6. Training and Testing the dataset:

Applying Random Forest Classifier

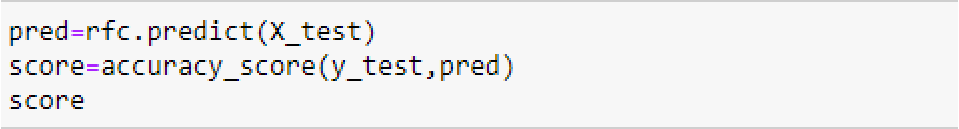


```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_jobs=3,oob_score=True,n_estimators=100,criterion="entropy")
model=rfc.fit(X_train,y_train)
```

```
print(X_test)
```

Prediction:

Prediction refers to the output of an algorithm after it has been trained on adataset.

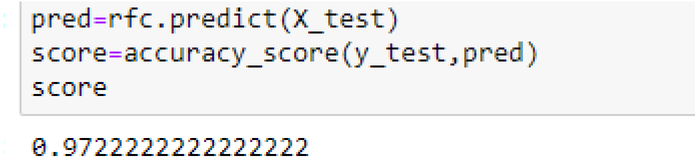


```
pred=rfc.predict(X_test)
score=accuracy_score(y_test,pred)
score
```

Fig.5.2.7. Prediction

Accuracy:

Accuracy is one metric for evaluating classification models. It is the Number of correct predictions. Accuracy = TP+TN/TP+FP+FN+TN.



```
pred=rfc.predict(X_test)
score=accuracy_score(y_test,pred)
score
```

```
0.9722222222222222
```

Fig.5.2.8. Accuracy

Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Precision = TP/TP+FP.

Recall (Sensitivity)

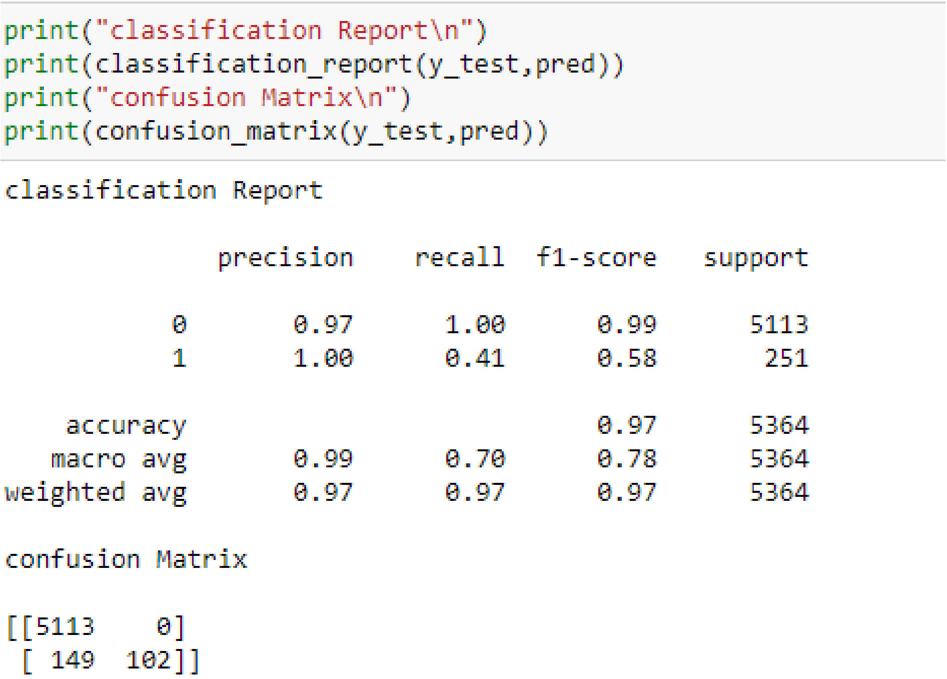
Recall is the ratio of correctly predicted positive observations to the all observations in actual class

Recall = TP/TP+FN.

F1 score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precisionand Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision).



```
print("classification Report\n")
print(classification_report(y_test,pred))
print("confusion Matrix\n")
print(confusion_matrix(y_test,pred))
```

```
classification Report

              precision    recall  f1-score   support

           0       0.97      1.00      0.99      5113
           1       1.00      0.41      0.58       251

    accuracy                           0.97      5364
   macro avg       0.99      0.70      0.78      5364
weighted avg       0.97      0.97      0.97      5364

confusion Matrix

[[5113    0]
 [ 149  102]]
```

Fig.5.2.9. Confusion Matrix

CHAPTER 6

TESTING

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we cansay,Testing is a process of executing a program with the intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error,if it exists.

The first approach is what known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

Black-Box Testing

This technique of testing is done without any knowledge of the interior workings of the application The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon.

- Well suited and efficient for large code segments.
- Code access is not required.
- Clearly separates user's perspectives from the developer's perspective through visibly defined roles.

White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. It is also called “glass testing” or “open-box testing”. In order to perform white- box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately.

In this, the test cases are generated on the logic of each module. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent modules have been executed.
- Execute all logical decisions and loops.
- Execute through proper plots and curves.

Performance Evaluation

This project has been successfully executed its source code. Initially there were some errors in the code. By resolving them, the code is fully free from errors and bugs.

After performing feature selection models are build using different classifiers. The classifier that gives best possible accuracy has been considered in this project. The classifier used to build a model in this project is Random forest classifier which is gave the effective results. Hence, for the considered dataset, the Random forest classifier performed well with the accuracy of 97.22.

CONCLUSION

Fake job postings are an important real-world challenge that require active solutions. This project aims to provide a potential solution to this problem. The textual data is pre-processed to generate optimal results and relevant numerical fields are chose as well. The output of Multiple models is combined to produce the best possible results. This is done to reduce the bias that a machine learning model has towards the dominant class.

The most interesting part of this project was how certain locations are an epitome of fraudulent jobs. For example, Bakersfield, California has a fake to real job ratio of 15:1. Places like this require some extra monitoring. Another interesting part was that most entry level jobs seem to be fraudulent. It seems like scammers tend to target younger people who have a bachelor’s degree or high school diploma looking for full-time jobs. The most challenging part was text datapreprocessing. The data was in a very format. Cleaning it required a lot of effort.

REFERENCES

[1] Fake Job Prediction Prediction Based on Prior Knowledge by www.hcup-us.ahrq.gov/nisoverview.jsp.

[2] GDPS-General Fake Job Prediction Prediction System by www.irjet.net.

[3] Fake Job Prediction Prediction Using Machine Learning by International Research Journal of Engineering and Technology (IRJET).

[4] Machine Learning Methods Used in Fake Job Prediction by www.wikipedia.com.

[5] https://www.researchgate.net/publication/325116774_FakeJobPrediction_prediction_usin g_machine_learning_techniques.

[6] https://www.slideshare.com/Fake Job Prediction_prediction.

[7] https:/ en.wikipedia.org /machine_learning_algorithms.

[8] https://en.wikipedia.org/wiki/Python_(programming_language).