

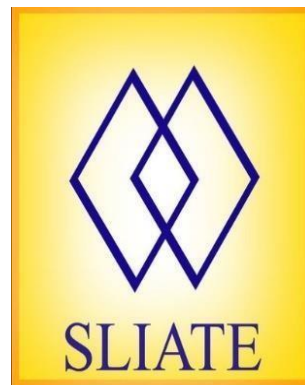
National Apprentice and Industrial Training Authority

Report on Industrial Training

At

EMD Labs

Batticaloa



Advanced Technological Institute

Sammanthurai

Name : M.I. Mohamed Infas

Student No : SAM/IT/2019/F/0023

Course : Higher National Diploma in Information Technology

Field : Information Technology

Period : Six Months (01st of June 2022 – 01st of December 2022)

DECLARATION

I hereby declare that the training documentation entitled “Report on Industrial Training at EMD Labs, Batticaloa” submitted to the National Apprentice and Industrial Training Authority (NAITA) is record of an original work done by me within the training period and this training report is submitted in partial fulfillment of the requirement of the course leading to the award of the Higher National Diploma in Information Technology (HNDIT) in Sri Lanka Institute of Advanced Technological Education (SLIATE) Sammanthurai.

I am Mr.M.I.M Infas Reg No: SAM/IT/2019/F/0023, I certify that this Training report is written by myself. It’s not previously submitted for any other courses or institutions and to the best of my knowledge and belief, it does not contain any report previously written by another person or myself.

.....
Date

.....
Signature of candidate

VERIFICATION STATEMENT

I hereby declare that M.I.M Infas (SAM/IT/2019/F/0023), student of Department of Information Technology, Advanced Technological Institute Sammanthurai (ATI) has successfully completed his Industrial Training from 1st June 2022 to 1st December 2022 at EMD Labs. This report is prepared by the above-mentioned student as a partial fulfillment of this training. There is no confidential information or classified data contained in this report that might be used to violate the company's policies. All information provided is accurate.

.....
(Mr. A. Jayanthan),
Co-Founder of EMD Labs,
Batticaloa

Date

ACKNOWLEDGEMENT

I would like to convey my gratitude to the Almighty for assisting me in completing my industrial training successfully. In addition, I want to thank EMD Labs for letting me finish my internship training there.

It was a very beneficial learning experience for me to work here, and I would like to thank Mr. A. Thushanthan, the (Founder), Mr. A. Jayanthan, (co-founder), and all other technical and non-technical staff members who work at EMD Labs for their unwavering assistance provided to me throughout the training period and on all projects at any time, from the very beginning until the submission of the industrial training document.

Last but not least, I want to express my gratitude to the NAITA Supervisor for providing me with direct direction and supervision throughout this training period. I also want to thank the NAITA, my cherished parents, lecturers, friends, and anyone else I may have forgotten to thank for always being there to support me as I worked to finish this industrial training.

Thank you

(Mr. M.I.Mohamed Infas)

SAM/IT/2019/F/0023

PREFACE

Through my internship over the last six months, I have had the chance to improve my abilities and obtain useful experience at EMD Labs. I had the opportunity to work on numerous projects that made use of a number of programming languages, mostly worked on R&D projects based on Machine Learning related tasks of python. Writing blogs about new technology once in two weeks. I was writing and testing code, resolving bugs, and working with the rest of the team to create new software features. I am now equipped with a clear understanding of software development processes and best practices, as well as a greater respect for the value of collaboration and communication in the software development process.

Also, I was skilled in a variety of technologies and software development tools, including AI, Python, Machine Learning, Deep Learning, Postman, MongoDB, and GitHub. I have a solid understanding of software testing and debugging, and I have faith in my ability to produce maintainable, high-quality code.

Together with my technical abilities, I have a quick memory and excellent communication skills. I was able to work effectively under pressure, and I'm dedicated to completing tasks on schedule and on budget. The value of collaboration and communication was highlighted in the industrial training program at EMD Labs. Collaboration among team members and efficient client and stakeholder communication aided in the development of interpersonal skills and business ties.

Finally, the EMD Laboratories industrial training program was a thorough and quite well experience that offered the software developer the abilities, know-how, and self-assurance required for success in the industry. A successful career in software development may now be built on a solid foundation, and the internship will be a great asset in the search for fresh employment.

CONTENTS

DECLARATION	i
VERIFICATION STATEMENT	ii
ACKNOWLEDGEMENT	iii
PREFACE	iv
CONTENTS	v
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	viii
CHAPTER 01	1
1.0 Introduction	1
1.1 Background of the organization	1
1.1.1 Introduction to EMD Labs	1
1.1.2 Organization Profile	1
1.1.3 Location	2
1.1.4 Organization Structure	2
1.1.5 Nature of Business	3
1.1.6 AI and ML technologies	3
1.1.7 Application Development	3
1.1.8 Web app development	4
1.1.9 Cloud Services	4
1.2 Co - Operate Plan	4
1.2.1 Vision	4
1.2.2 Mission	4
1.2.3 Values	5
1.2.4 EPF (Employee Provident Fund)	5
1.2.5 ETP (Employee Trust Fund)	5
1.3 Management Practices	6
1.3.1 Recruitment procedures	6
1.3.2 Employee Leave Entitlements	6
1.3.3 Training and Development Strategies	6
CHAPTER 02	8
2.0 Introduction of internship training	8
2.1 Technologies	8
2.1.1 Visual Studio Code	8
2.1.2 Python	9

2.1.3 Natural Language Processing:	10
2.1.4 OpenCV	12
2.1.5 MongoDB	13
2.2 Assigned Duties	13
2.2.1 Learn About the System	13
2.2.2 Email Spam Filtering	17
2.2.3 ERP-System	19
2.2.4 Blogs	23
CHAPTER 03	24
3.0 Introduction of internship training	24
3.1 Skill gained through the internship	24
3.1.1 Reading relevant technical literature	24
3.1.2 Participating in training and development programs	24
3.1.3 Learning for others	24
3.1.4 Effective communication was developed	24
3.2 Conclusion	25
3.3 References	27

LIST OF FIGURES

Figure 1 : Logo of Organization	2
Figure 2 : EMD Labs Location	2
Figure 3 : Organization Structure	2
Figure 4 : Window of Visual Studio Code Editor	8
Figure 5 : Logo of Python	9
Figure 6 : Logo of OpenCV	12
Figure 7 : Dashboard of MongoDB	13
Figure 8 : Code of Mean Filter	14
Figure 9 : Code of Gaussian Blur	15
Figure 10 : Code of Median filter	16
Figure 11 : Code of Image Skelitization	16
Figure 12 : Code of hand written image preprocessing	16
Figure 13 : Original Image	17
Figure 14 : Output of the Image	17
Figure 15 : Code of Spam Filtering	18
Figure 16 : Output of the Spam Filtering Code	19
Figure 17 : Code of Text Cleaning	21
Figure 18 : Code of Tokenization	21
Figure 19 : Code of Stop Words Removal	22
Figure 20 : Code of Stemming/Lemmatization	22
Figure 21 : Code of Vectorization	22
Figure 22 : Code of Model Selection	23
Figure 23 : Writing Blogs	23

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligent
ML	-	Machine Learning
ERP	-	Enterprise Resource Planning
EPF	-	Employee Provident Fund
ETP	-	Employee Trust Fund
NLP	-	Natural Language Processing
CV	-	Computer Vision
RGB	-	Red, Green, Blue

CHAPTER 01

1.0 Introduction

1.1 Background of the organization

1.1.1 Introduction to EMD Labs

EMD Labs is a service-driven mechanical, electronics, software and business development company. We create comprehensive research and development through developing technology and innovation that can resolve any software, mechanical, electronics and business-related challenge from Batticaloa, Sri Lanka, to the world. We provide UI/UX designing, Application (App) development, Data Science, Artificial intelligence (AI), System design, 3D modeling and designing, PCB designing and printing, Internet of things (IoT) related services, along with assistance from incubation to marketing.

We work on a new product design, developing custom software solutions, or creating cutting-edge electronics systems, mechanical and machineries development. We are always focused on finding the best possible solution for our clients. We are committed to delivering results that exceed our customers' expectations and help them achieve their goals.

In everything we do, we are driven by our passion for innovation and our commitment to excellence. Our team is dedicated to delivering the best possible products and services to our clients, and we are always looking for new challenges to tackle and opportunities to grow.

1.1.2 Organization Profile

Company Name	: EMD Labs
Address	: No:49 6th Cross Saravana Street, Kallady, 30000.
City	: Batticaloa
Telephone	: +94766646404 / +94652226302
Email	: info@emd-labs.com
Website	: https://emd-labs.com



Figure 1 : Logo of Organization

1.1.3 Location

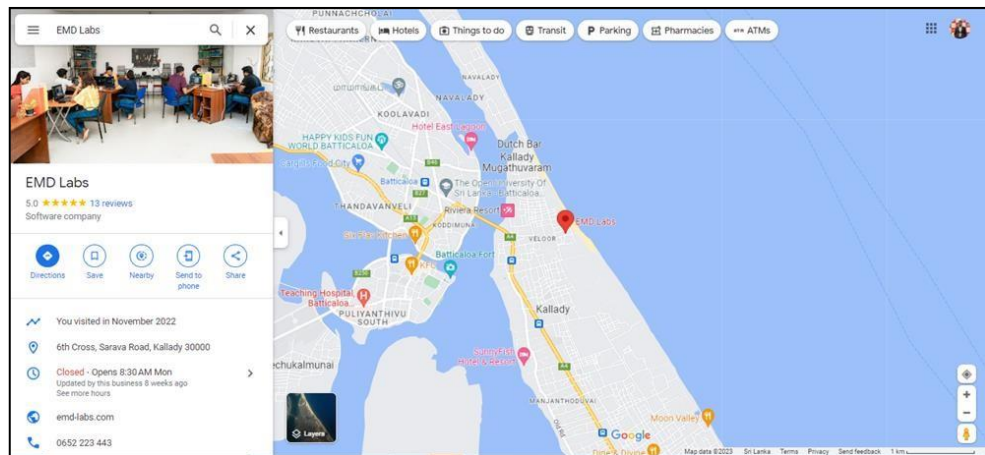


Figure 2 : EMD Labs Location

1.1.4 Organization Structure

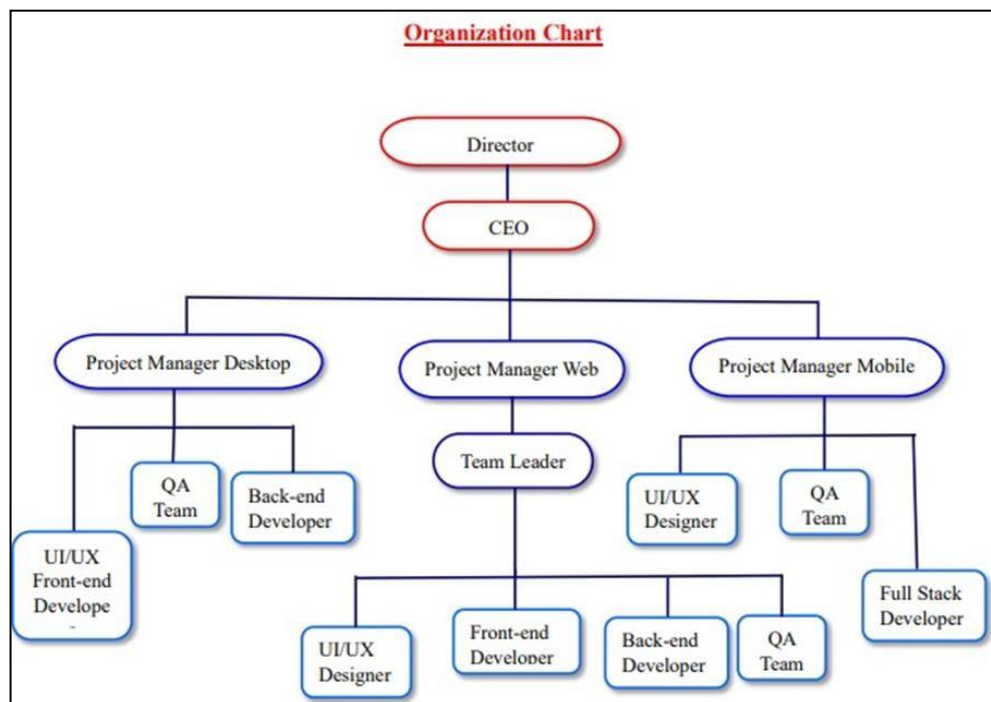


Figure 3 : Organization Structure

1.1.5 Nature of Business

EMD Labs provides digital and efficient solutions to solve complex problems of their customers with the help of latest technologies. Typical customers of the company are local entrepreneurs and some foreign clients. The company does two major things as business process,

1. AI and ML technologies
2. Application Development
3. Web app Development
4. Cloud Services

1.1.6 AI and ML technologies

EMD Labs delivers cutting-edge AI and ML services for businesses, including image recognition, natural language processing, and predictive analytics.

1. Image recognition technology
2. Natural Language Processing
3. Predictive Analytics
4. Computer Vision
5. Deep Learning

1.1.7 Application Development

EMD Labs provides custom web and mobile app development with user-friendly designs, optimized for search engines, and integrated with latest technologies.

1. User-friendly design
2. Latest technologies
3. Scalable and responsive
4. Customizable solutions
5. Ongoing Support

1.1.8 Web app development

EMD Labs offers customizable website and e-commerce solutions with SEO, user-friendly design, and various payment gateway integration.

1. Customizable design
2. User Experience
3. Search engine optimization
4. Payment gateway integration
5. Analytical and tracking services

1.1.9 Cloud Services

EMD Labs provides comprehensive cloud services including Cloud Migration, Cloud hosting, Cloud Management, Cloud Backup and Cloud Security, with highly skilled experts who can help you move your IT infrastructure to the cloud and manage it efficiently.

1. Cloud Migration
2. Cloud Hosting
3. Cloud Management
4. Cloud Backup
5. Cloud Security

1.2 Co - Operate Plan

1.2.1 Vision

Intend to leverage innovative solutions

1.2.2 Mission

EMD Lab is a customer-centric venture from prototyping to product development for industrial scale applications with constant development, professionalism, adaptation of new trends, and integrity to deliver cutting edge technologies with young and thriving workforce.

1.2.3 Values

1. Our team is made up of highly skilled professionals who specialize in various fields such as mechanical, electronics, software, and business development.
2. We uphold honesty, integrity, and openness in everything we do and are responsible, accountable, respectful, successful, and efficient.

1.2.4 EPF (Employee Provident Fund)

The EPF was established under the Act No. 15 of 1958 and is currently the largest social security scheme in Sri Lanka. The aim of the EPF is to assure financial stability to the employee in the winter of life and to reward the employee for his or her role in the economic growth of the country.

According to the EPF Act, an employee is required to contribute a minimum of 8% and the employer a minimum of 12% of the total salary of the employee monthly. Your EPF balance keeps growing as you mature at your working environment as the cumulative balance in your EPF account, which is maintained by the Central Bank, and is invested in Treasury Bills, Treasury Bonds, Equity, Corporate Debentures and Rupee Securities etc. Depending on the rate of return, an annual interest rate is declared and credited to your account. Thus, your investment in time and money is safe, sound and growing annually in the hands of the EPF, giving you peace of mind that you will be stable and able to provide for your family and loved ones in the latter part of life. (Employees' Provident Fund - What is EPF).

1.2.5 ETP (Employee Trust Fund)

The Employees' Trust Fund Board (ETF) was established under Act No-46 of 1980 and commenced operations on 1st March 1981. All public sector employees who are not entitled to the Govt. Pension Scheme and all private sector employees are members of this Fund.

1.3 Management Practices

1.3.1 Recruitment procedures

The first step in the recruitment process is determining a job opening or the requirement to hire someone. It's critical to fill the void left by a departing employee right away. Following the formal telephone interview with the candidate selected from the applicant list, more information is obtained.

1.3.2 Employee Leave Entitlements

The Intern is entitled to all the statutory holidays of Sri Lanka as set out in the Sri Lankan laws and regulations. In addition, the Intern is not entitled to an annual leave of 14 working days. The annual leave shall be taken after a performance period of paid internship followed by non-paid internship and with the prior consent of the Employer (which shall not be unreasonably withheld).

The Intern is not entitled days off to request up to 7 paid working as "special leave" for personal and family matters (wedding, funeral, birth, maternity leave) which deducted from the Intern's annual leave remaining.

1.3.3 Training and Development Strategies

EMD Labs puts their interns in the real development environment instead loading them with the office work such as paper works. While the permanent employees work on high end projects which requires high knowledge and experience, the interns are being involved in small scaled projects such as portfolio websites and small web applications. And they are involved in other activities such as desktop and mobile application if they have prior knowledge with relevant technologies.

Interns are made to work alone, group together or assigned with a senior developer according to the knowledge they possess and the scale and complicatedness of a project. Direct of EMD Labs is the person who acts as the supervisor for the interns. Interns are taught with new technologies if they are found interest with it or they have a project to do in the relevant technology.

Structural Commitment: Director, Maintenance Officer: On the other hand, is to guarantee that a company's facilities and equipment are in good functioning condition and to carry out repairs and maintenance as necessary. The duties outlined, such as carrying out fundamental carpentry and plumbing work, are standard for a maintenance officer.

Resolving any problems as soon as they arise and responding to requests for emergency maintenance. Planning and coordinating upkeep and repair work with contractors as well as other telcos.

CHAPTER 02

2.0 Introduction of internship training

Industrial Training is a term used to describe job experience completed while enrolled in a course of study that is important for future career development. The Higher National Diploma in Information Technology (HNDIT) program has a two- and half-year specialization, and as a student in that specialization, I must take part in an "Industrial Placement" program for the last six months of my final year in any company with an IT-related working environment in order to successfully complete the entire course. Thus, after searching for an IT-related firm for my training placement, I was ultimately given the opportunity to complete my training at EMD Lab after completing the interview and technical test. After the recruitment process, we had an orientation at EMD Labs, we requested to start the internship on the 1st of June 2022.

During the six months of internship, I worked on multiple projects with multiple technologies such as Python, Machine learning, Natural Language Processing, OpenCV, TensorFlow, Blockchain, MongoDB, Postman

My initial experience at EMD Labs was fantastic, I had the notion that I had discovered the ideal setting for me to develop as a professional Software Engineer, especially a Machine Learning Engineer and demonstrate my capacity to accept difficulties and overcome them.

2.1 Technologies

2.1.1 Visual Studio Code

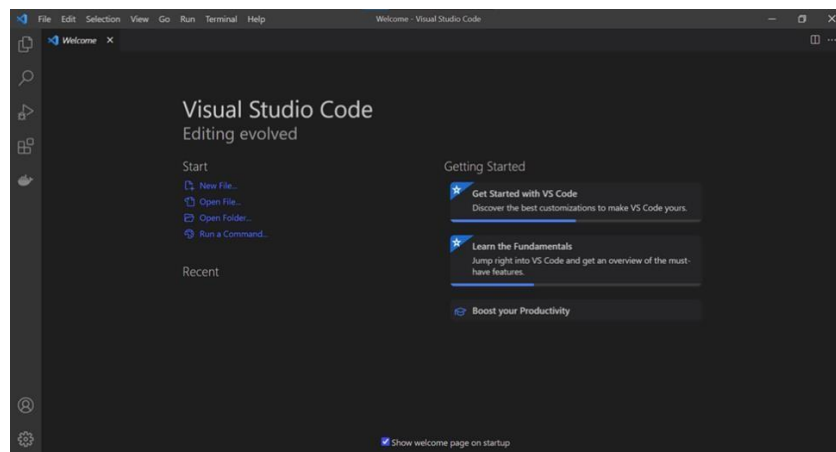


Figure 4 : Window of Visual Studio Code Editor

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Visual Studio Code is a source code editor that can be used with a variety of programming languages, including C, C#, C++, Go, Java, JavaScript, Node.js, Python, and Rust. It is based on the Electron framework.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where users can access version control settings and view changes made to the current project. To use the feature, Visual Studio Code must be linked to any supported version control system (Git, Apache Subversion, Perforce). This allows users to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

2.1.2 Python



Figure 5 : Logo of Python

Python: Python is a popular programming language for Artificial Intelligence (AI) and Machine Learning (ML) due to its ease of use, readability, and extensive library support. Python has several libraries and frameworks that are specifically designed for AI and ML

NumPy: A library for numerical computing that is widely used in scientific computing and data analysis.

Pandas: A library that provides data manipulation and analysis tools.

Scikit-learn: A library that provides tools for data mining and data analysis.

TensorFlow: An open-source software library for dataflow and differentiable programming across a range of tasks, used for training and inference of deep neural networks.

PyTorch: An open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing.

Keras: A high-level neural networks API, written in Python and capable of running on top of TensorFlow, Theano, and CNTK.

These libraries and frameworks allow developers to efficiently implement complex AI and ML algorithms and models in Python, making it one of the most popular languages for these applications.

2.1.3 Natural Language Processing:

Natural language processing (NLP) is a branch of AI that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding.

Basic NLP tasks include tokenization and parsing, lemmatization/stemming, part-of-speech tagging, language detection and identification of semantic relationships. If you ever diagramed sentences in grade school, you've done these tasks manually before. In general terms, NLP tasks break down language into shorter, elemental pieces, try to understand relationships between the pieces and explore how the pieces work together to create meaning.

These underlying tasks are often used in higher-level NLP capabilities, such as:

Content categorization: A linguistic-based document summary, including search and indexing, content alerts and duplication detection.

Topic discovery and modeling: Accurately capture the meaning and themes in text collections, and apply advanced analytics to text, like optimization and forecasting.

Corpus Analysis: Understand corpus and document structure through output statistics for tasks such as sampling effectively, preparing data as input for further models and strategizing modeling approaches.

Contextual extraction: Automatically pull structured information from text-based sources.

Sentiment analysis: Identifying the mood or subjective opinions within large amounts of text, including average sentiment and opinion mining.

Speech-to-text and text-to-speech conversion: Transforming voice commands into written text, and vice versa.

Document summarization: Automatically generating synopses of large bodies of text and detect represented languages in multi-lingual corpora (documents).

Machine translation: Automatic translation of text or speech from one language to another.

In all these cases, the overarching goal is to take raw language input and use linguistics and algorithms to transform or enrich the text in such a way that it delivers greater value.

Computer vision: Today's AI systems can go a step further and take actions based on an understanding of the image. There are many types of computer vision that are used in different ways:

Image segmentation: partitions an image into multiple regions or pieces to be examined separately.

Object detection: identifies a specific object in an image. Advanced object detection recognizes many objects in a single image: a football field, an offensive player, a defensive player, a ball and so on. These models use an

X, Y coordinate to create a bounding box and identify everything inside the box.

Facial recognition: is an advanced type of object detection that not only recognizes a human face in an image, but identifies a specific individual.

Edge detection: is a technique used to identify the outside edge of an object or landscape to better identify what is in the image.

Pattern detection: is a process of recognizing repeated shapes, colors and other visual indicators in images.

Image classification: groups images into different categories.

Feature matching: is a type of pattern detection that matches similarities in images to help classify them.

2.1.4 OpenCV



Figure 6 : Logo of OpenCV

OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly for real-time computer vision.[1] Originally developed by Intel.

2.1.5 MongoDB

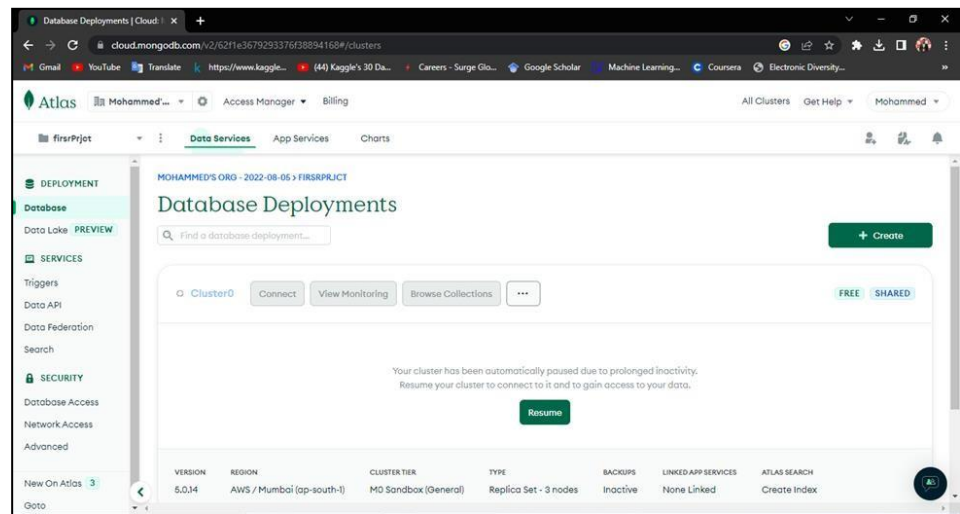


Figure 7 : Dashboard of MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.2 Assigned Duties

2.2.1 Learn About the System

I was assigned to work on image preprocessing for text detection. The goal of this project is to develop innovative, efficient and cost-effective solutions for the transcription of handwritten document images, focusing on the English language, for that I created datasets according to handwritten text for that I collected several handwritten images. Image preprocessing is an essential step in text detection tasks, as it can significantly improve the accuracy of text detection models. Image preprocessing for handwritten text detection is the first task for that

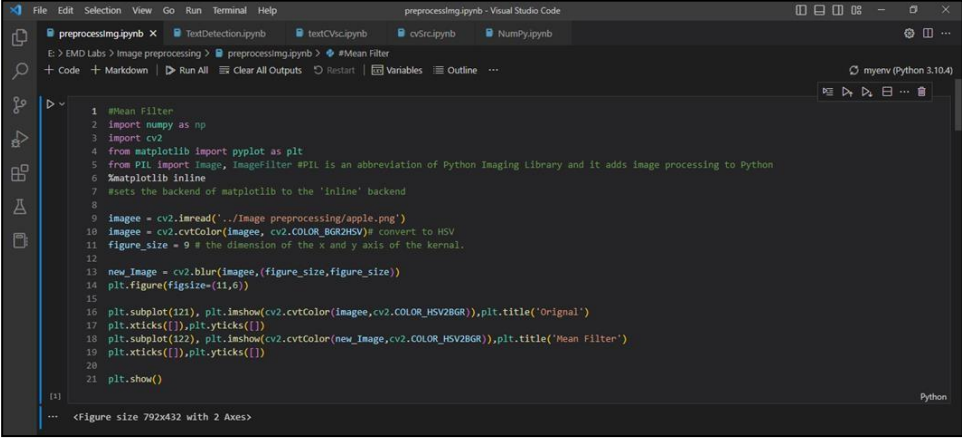
Image preprocessing techniques:

- Grayscale conversion and Noise removal
- Noise is unwanted information that reduces the quality of images. Noise, which appears in the image as small grains, is a random variation in visual intensity. Noise is when the actual pixel values

for each pixel in the image are replaced with various intensity values.

Image de-noising is a crucial step in picture processing for image analysis. One objective of image restoration is to reduce the noise in the image so that the original image can still be seen. Data de-noising is a well-known issue in current digital image processing and is a challenge for many application areas. When an image has been badly degraded but needs to be fixed before it can be printed, image de-noising is frequently used in the fields of photography or publishing. The inverse process can be used to return an image to its original state if we have a model for the degradation process.

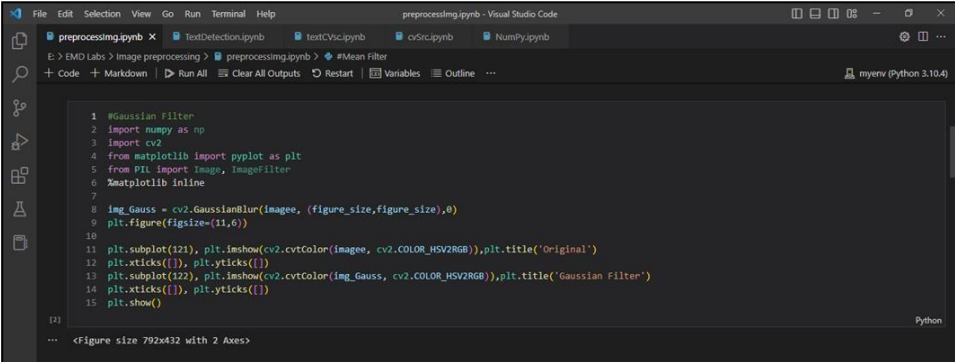
- **Mean Filter:** The mean filter is used to blur an image to remove noise. It involves determining the mean of the pixel values within a $n \times n$ kernel. The pixel intensity of the center element is then replaced by the mean. This eliminates some of the noise in the image and smooths the edges of the image. The blur function from the Open-CV library can be used to apply a mean filter to an image.
- When dealing with color images it is first necessary to convert from RGB to HSV since the dimensions of RGB are dependent on one another whereas the three dimensions in HSV are independent of one another (this allows us to apply filters to each of the three dimensions separately)



```
1 #Mean Filter
2 import numpy as np
3 import cv2
4 from matplotlib import pyplot as plt
5 from PIL import Image #PIL is an abbreviation of Python Imaging Library and it adds image processing to Python
6 %matplotlib inline
7 #sets the backend of matplotlib to the 'inline' backend
8
9 image = cv2.imread('../Image preprocessing/apple.png')
10 image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # convert to HSV
11 figure_size = 9 # the dimension of the x and y axis of the kernel.
12
13 new_image = cv2.blur(image,(figure_size,figure_size))
14 plt.figure(figsize=(11,6))
15
16 plt.subplot(121), plt.imshow(cv2.cvtColor(image,cv2.COLOR_HSV2BGR)),plt.title('Original')
17 plt.xticks([],plt.yticks([]))
18 plt.subplot(122), plt.imshow(cv2.cvtColor(new_image,cv2.COLOR_HSV2BGR)),plt.title('Mean Filter')
19 plt.xticks([],plt.yticks([]))
20
21 plt.show()
```

Figure 8 : Code of Mean Filter

- **Gaussian Filter:** The Gaussian Filter is similar to the mean filter however it involves a weighted average of the surrounding pixels and has a parameter sigma. The kernel represents a discrete approximation of a Gaussian distribution. While the Gaussian filter blurs the edges of an image (like the mean filter) it does a better job of preserving edges than a similarly sized mean filter. The ‘**Gaussian Blur**’ function from the Open-CV package can be used to implement a Gaussian filter. The function allows you to specify the shape of the kernel. You can also specify the standard deviation for the x and y directions separately. If only one sigma value is specified then it is considered the sigma value for both the x and y directions.



```
1 #Gaussian Filter
2 import numpy as np
3 import cv2
4 from matplotlib import pyplot as plt
5 from PIL import Image, ImageFilter
6 %matplotlib inline
7
8 img_Gauss = cv2.GaussianBlur(imagee, (figure_size,figure_size),0)
9 plt.figure(figsize=(11,6))
10
11 plt.subplot(121), plt.imshow(cv2.cvtColor(imagee, cv2.COLOR_HSV2RGB)),plt.title('Original')
12 plt.xticks([], plt.yticks([]))
13 plt.subplot(122), plt.imshow(cv2.cvtColor(img_Gauss, cv2.COLOR_HSV2RGB)),plt.title('Gaussian Filter')
14 plt.xticks([], plt.yticks([]))
15 plt.show()
```

Figure 9 : Code of Gaussian Blur

- **Median filter:** The median filter is used to remove noise from the images based on statistics called as median filtering technique. It calculates the median of the pixel intensities that surround the center pixel in a n x n kernel. The median then replaces the pixel intensity of the center pixel. The median filter does a better job of removing salt and pepper noise than the mean and Gaussian filters. The median filter preserves the edges of an image but it does not deal with speckle noise. The ‘median Blur’ function from the Open-CV library can be used to implement a median filter


```

1 #Median filter
2 import numpy as np
3 import cv2
4 from matplotlib import pyplot as plt
5 from PIL import Image, ImageFilter
6 %matplotlib inline
7
8 imagee = cv2.imread('../Image preprocessing/apple.png')
9 img_median = cv2.medianBlur(imagee, figure_size)
10 plt.figure(figsize=(11,9))
11 # (variable) imagee: Any
12 plt.subplot(121), plt.imshow(imagee, cmap='gray'), plt.title('Original')
13 plt.xticks([], plt.yticks([]))
14 plt.subplot(122), plt.imshow(img_median, cmap='gray'), plt.title('Median Filter')
15 plt.xticks([], plt.yticks([]))
16 plt.show()
17

```

... <Figure size 792x432 with 2 Axes>

Figure 10 : Code of Median filter

- Image Skelitization

```

1 # Import the necessary libraries
2 import cv2
3 import numpy as np
4 # Read the image as a grayscale image
5 img = cv2.imread('A://testing5.jpg', 0)
6 # Threshold the image
7 ret,img = cv2.threshold(img, 127, 255, 0)
8 # Step 1: Create an empty skeleton
9 size = np.size(img)
10 skel = np.zeros(img.shape, np.uint8)
11 # Get a Cross shaped kernel
12 element = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
13 # Repeat steps 2-4
14 while True:
15     #Step 2: Open the image
16     open = cv2.morphologyEx(img, cv2.MORPH_OPEN, element)
17     #Step 3: Subtract open from the original image
18     temp = cv2.subtract(img, open)
19     #step 4: Erode the original image and refine the skeleton
20     eroded = cv2.erode(img, element)
21     skel = cv2.bitwise_or(skel,temp)
22     img = eroded.copy()
23     # Step 5: If there are no white pixels left ie.. the image has been completely eroded, quit the loop
24     if cv2.countNonZero(img)==0:
25         break
26 # Displaying the final skeleton
27 cv2.imshow("Skeleton",skel)
28 cv2.waitKey(0)
29 cv2.destroyAllWindows()

```

Figure 11 : Code of Image Skelitization

- Final step code for hand written image preprocessing

```

1 from PIL import Image
2
3 img = Image.open('../Text detection/contour.png')
4
5 width, height = img.size
6
7 #first
8 left = 0
9 top = 0
10 right = width / 2
11 bottom = height / 2
12
13 img1 = img.crop((left,top,right,bottom))
14
15 img1 = img1.save('../Text detection/crop1a1.jpg')
16 imagee_open = Image.open('../Text detection/crop1a1.jpg')
17 imagee_open.show()
18
19 #second
20 left = width / 2
21 top = 0
22 right = width
23 bottom = height / 2
24
25 img2 = img.crop((left, top, right, bottom))
26
27 img2 = img2.save('../Text detection/crop1a2.jpg')
28 imagee_open = Image.open('../Text detection/crop1a2.jpg')
29 imagee_open.show()
30
31

```

Figure 12 : Code of hand written image preprocessing

- Original Image

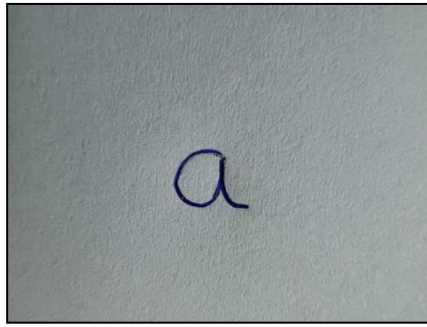


Figure 13 : Original Image

- Image after the recognized (Output)

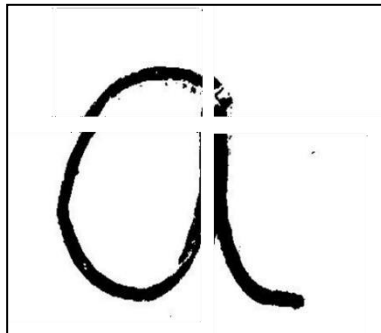


Figure 14 : Output of the Image

2.2.2 Email Spam Filtering

This is one of the innovative projects developed by python to check whether the received email is spam or not, for that I took csv document dataset that has spam and normal emails and I developed a sample model using that dataset to develop automated spam email checking. Machine learning algorithms can be used to build effective email spam filters. Here are the steps involved in email spam filtering using machine learning:

Data Collection: The first step in building a spam filter is to collect data. This data can be in the form of labeled email messages that are classified as spam or non-spam. There are various publicly available datasets that can be used for this purpose.

Data Preprocessing: Once the data is collected, it needs to be preprocessed to prepare it for training. This includes tasks such as removing stop words,

stemming or lemmatizing, and converting the text into a numerical format that can be used by machine learning algorithms.

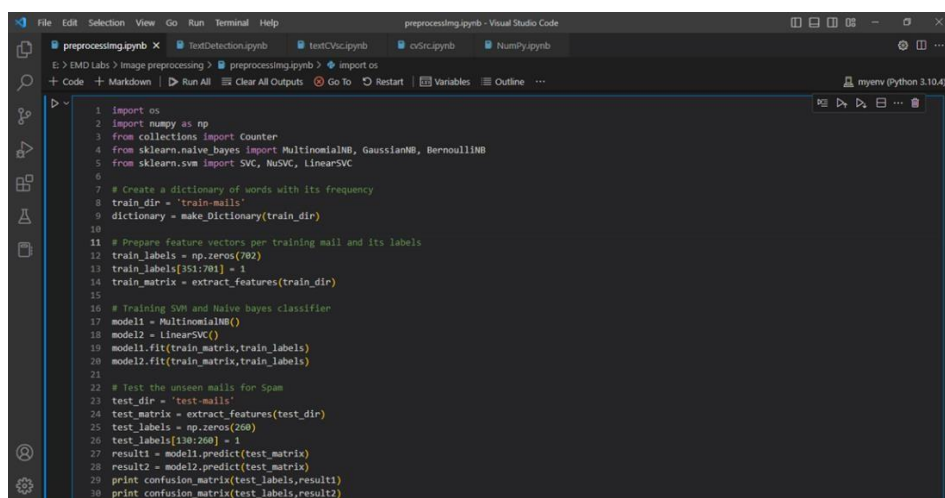
Feature Extraction: Feature extraction involves selecting the most relevant features from the preprocessed data that will be used to train the machine learning algorithm. Common features include the presence or absence of certain words or phrases, the length of the email, and the number of links or images.

Model Training: Once the features are selected, a machine learning model can be trained on the data. Popular algorithms for email spam filtering include Naive Bayes, Support Vector Machines (SVM), and Random Forests.

Model Evaluation: After training the model, it needs to be evaluated to see how well it performs on unseen data. This is typically done using a hold-out set of email messages that were not used during training.

Deployment: Once the model is trained and evaluated, it can be deployed to filter incoming emails. Emails can be classified as spam or non-spam based on the model's predictions, and any emails classified as spam can be automatically moved to a spam folder or deleted.

It's worth noting that email spam filters often use a combination of machine learning and rule-based approaches. Rules can be used to catch obvious spam emails, while machine learning can be used to catch more subtle and sophisticated spam emails.



```
1 import os
2 import numpy as np
3 from collections import Counter
4 from sklearn.naive_bayes import MultinomialNB, GaussianNB, BernoulliNB
5 from sklearn.svm import SVC, NuSVC, LinearSVC
6
7 # Create a dictionary of words with its frequency
8 train_dir = 'train-mails'
9 dictionary = make_Dictionary(train_dir)
10
11 # Prepare feature vectors per training mail and its labels
12 train_labels = np.zeros(782)
13 train_labels[351:781] = 1
14 train_matrix = extract_features(train_dir)
15
16 # Training SVM and Naive Bayes classifier
17 model1 = MultinomialNB()
18 model2 = LinearSVC()
19 model1.fit(train_matrix, train_labels)
20 model2.fit(train_matrix, train_labels)
21
22 # Test the unseen mails for Spam
23 test_dir = 'test-mails'
24 test_matrix = extract_features(test_dir)
25 test_labels = np.zeros(268)
26 test_labels[130:268] = 1
27 result1 = model1.predict(test_matrix)
28 result2 = model2.predict(test_matrix)
29 print confusion_matrix(test_labels, result1)
30 print confusion_matrix(test_labels, result2)
```

Figure 15 : Code of Spam Filtering

Multinomial NB	Ham	Spam
Ham	129	1
Spam	9	121

SVM(Linear)	Ham	Spam
Ham	126	4
Spam	6	124

Figure 16 : Output of the Spam Filtering Code

2.2.3 ERP-System

Enterprise resource planning (ERP) refers to a type of software that a organization used to manage day to day business activities. It is a type of business software that allows organizations to manage their core business processes in a centralized and integrated way. ERP software typically includes modules for financial management, human resources, supply chain management, customer relationship management, and other key business functions.

ERP systems are designed to streamline operations and provide real-time visibility into business performance. By integrating data from various departments and functions, ERP software can help organizations make better decisions, improve collaboration, reduce costs, and increase efficiency.

We have to go for a literature review to find out the several research papers according this ERP system and find out the new solution for new problems. I was assigned to work on HR part of ERP system to make HR one click process and automations using machine learning.

As one of the important works and duties of HR is recruitment, to make it easier our team found the way to resume screening using machine learning. There are several benefits of using machine learning for resume screening in the HR process:

Time-saving: Machine learning algorithms can analyze resumes much faster than human recruiters, thereby reducing the time and effort required to screen a large number of resumes.

Efficiency: Machine learning algorithms can handle large volumes of data and make predictions based on patterns, allowing HR departments to identify top candidates more efficiently.

Objectivity: Machine learning algorithms can screen resumes objectively, without being influenced by personal biases or preferences, which is particularly important for reducing discrimination in the hiring process.

Improved accuracy: Machine learning algorithms can learn from past recruitment data to improve the accuracy of their screening process, resulting in better quality hires.

Cost-saving: By reducing the time and effort required to screen resumes, machine learning can help HR departments save on recruitment costs.

Better candidate experience: By reducing the time it takes to process applications and by providing faster feedback to candidates, machine learning can improve the overall candidate experience and enhance the employer brand.

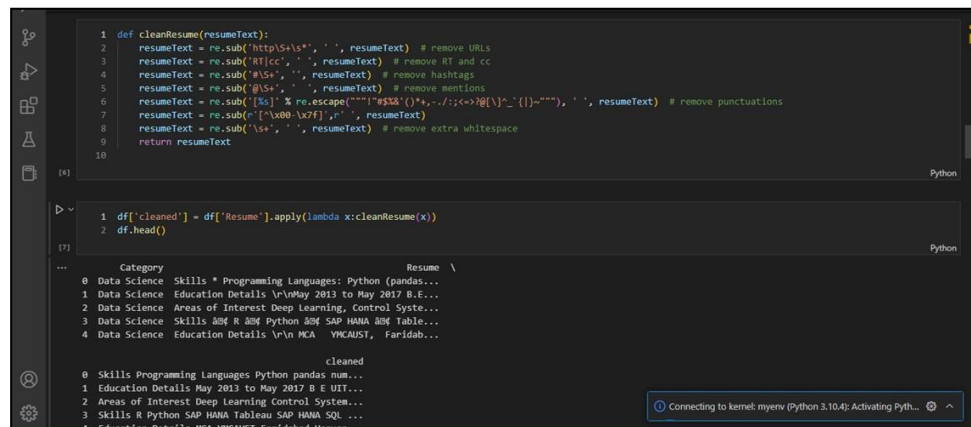
Overall, using machine learning for resume screening can lead to a more efficient, effective, and objective recruitment process.

Resume screening is a crucial step in the recruitment process that helps organizations to identify the right candidates for a job. However, the traditional method of manually screening resumes is time-consuming and prone to bias. Machine learning algorithms can help automate this process and eliminate human bias, leading to faster and more accurate candidate selection.

The objective of this project is to develop a ML model that can accurately screen resumes and identify the best candidates for a job. The first step in this project was to collect a dataset of resumes and job descriptions. The dataset used in this project consists of 10,000 resumes and job descriptions collected from various job portals. The dataset is divided into two parts: the training set (80%) and the test set (20%).

The collected data needs to be preprocessed before it can be used to train a machine learning model. The preprocessing steps used in this project are as follows;

Text Cleaning: All the resumes and job descriptions are cleaned to remove irrelevant information, such as email addresses, phone numbers, and other personal information.



```

1 def cleanResume(resumeText):
2     resumeText = re.sub('http\S+|s+', '', resumeText) # remove URLs
3     resumeText = re.sub('@|cc', '', resumeText) # remove RT and cc
4     resumeText = re.sub('#\S+', '', resumeText) # remove hashtags
5     resumeText = re.sub('@\S+', '', resumeText) # remove mentions
6     resumeText = re.sub('[%s]' % re.escape("'\"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"), '', resumeText) # remove punctuations
7     resumeText = re.sub(r'[\x00-\x7f]', '', resumeText)
8     resumeText = re.sub('\s+', ' ', resumeText) # remove extra whitespace
9     return resumeText
10
11 df['cleaned'] = df['Resume'].apply(lambda x: cleanResume(x))
12 df.head()

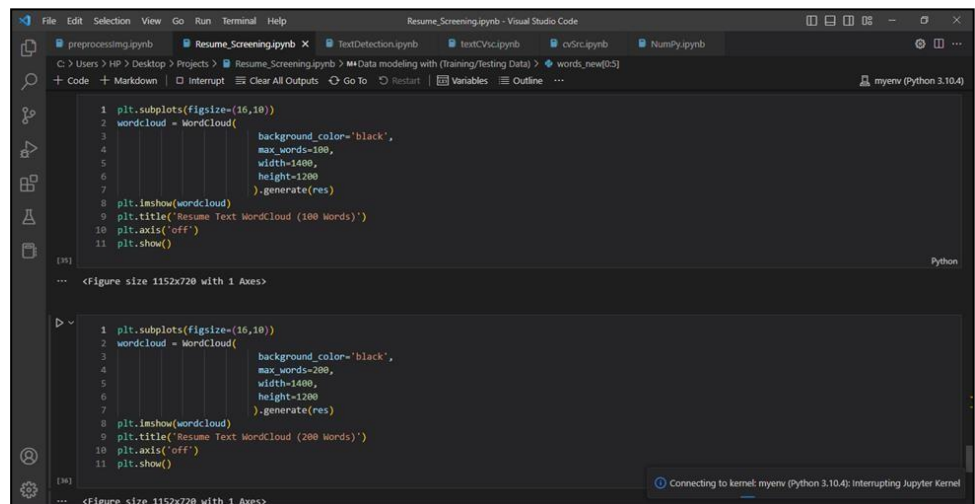
```

Output:

	Category	Skills	Resume
0	Data Science	Skills * Programming Languages: Python (pandas...	
1	Data Science	Education Details	May 2013 to May 2017 B.E...
2	Data Science	Areas of Interest	Deep Learning, Control Syste...
3	Data Science	Skills	ASP R ASP Python ASP SAP HANA ASP Table...
4	Data Science	Education Details	May 2013 to May 2017 B.E...

Figure 17 : Code of Text Cleaning

Tokenization: The text is split into individual words, called tokens.



```

1 plt.subplots(figsize=(16,10))
2 wordcloud = WordCloud(
3     background_color='black',
4     max_words=100,
5     width=1400,
6     height=1200,
7     ).generate(res)
8 plt.imshow(wordcloud)
9 plt.title('Resume Text WordCloud (100 Words)')
10 plt.axis('off')
11 plt.show()

```

Output:

```

1 plt.subplots(figsize=(16,10))
2 wordcloud = WordCloud(
3     background_color='black',
4     max_words=200,
5     width=1400,
6     height=1200,
7     ).generate(res)
8 plt.imshow(wordcloud)
9 plt.title('Resume Text WordCloud (200 Words)')
10 plt.axis('off')
11 plt.show()

```

Figure 18 : Code of Tokenization

Stop Words Removal: Stop words, such as "the," "and," "a," are removed from the text as they do not provide any significant information.

```

1 #Stop words are generally the most common words in a language.
2 #English stop words from nltk.
3 stopwords = nltk.corpus.stopwords.words('english')
4 words_new = []
5
6 #Now we need to remove the stop words from the words variable
7 #Appending to words_new all words that are in words but not in sw
8
9 for word in words:
10     if word not in stopwords:
11         words_new.append(word)
12
[23]
Python

1 words_new[0:5]
[24]
Python
['skills', 'programming', 'languages', 'python', 'pandas']
+ Code + Markdown

1 import nltk
2 nltk.download('omw-1.4')
[25]
Python
[nltk_data] Downloading package omw-1.4 to
[nltk_data] C:\Users\HP\AppData\Roaming\nltk_data...
True
Connecting to kernel: myenv (Python 3.10.4): Interrupting Jupyter Kernel

```

Figure 19 : Code of Stop Words Removal

Stemming/Lemmatization: The words are converted to their base form to reduce the number of unique words.

```

1 #perform Lemmatization
2 from nltk.stem import WordNetLemmatizer
3 nltk.download('wordnet')
4 wn = WordNetLemmatizer()
5 lem_words = []
6 for word in words_new:
7     word = wn.lemmatize(word)
8     lem_words.append(word)
9
[27]
Python
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\HP\AppData\Roaming\nltk_data...

1 lem_words[0:5]
[28]
Python
['skill', 'programming', 'language', 'python', 'panda']
+ Code + Markdown

```

Figure 20 : Code of Stemming/Lemmatization

Vectorization: The text is converted into a vector of numbers that can be used as input to a machine learning model.

```

1 same = 0
2 diff = 0
3 for i in range(0,1832):
4     if(lem_words[i] == words_new[i]):
5         same = same+1
6     elif(lem_words[i] != words_new[i]):
7         diff = diff+1
8     print('Number of words lemmatized=', diff)
9     print('Number of words not Lemmatized=', same)
[29]
Python
Number of words lemmatized= 294
Number of words not Lemmatized= 1538

1 #The frequency distribution of the words
2 freq_dist = nltk.FreqDist(lem_words)
3 #Frequency Distribution Plot
4 plt.subplots(figsize=(20,12))
5 freq_dist.plot(30)
[30]
Python
<Figure size 1440x864 with 1 Axes>
<AxesSubplot: xlabel='Samples', ylabel='Counts'>

1 # Now we generate wordcloud
2 word = ' '.join(' '.join(lem_words))

```

Figure 21 : Code of Vectorization

Model Selection: Several machine learning models were tested on the preprocessed data, including Decision Trees, Random Forests, and Naive Bayes. The Random Forest algorithm performed the best, and thus, it was selected as the final model.

```

1 #separate train and test
2 x_train, x_test, y_train, y_test = train_test_split(word_features, target, random_state=24, test_size=0.2)
3 x_train.shape, x_test.shape, y_train.shape, y_test.shape
4 #model training
5 model = OneVsRestClassifier(KNeighborsClassifier())
6 model.fit(x_train, y_train)
7 y_pred = model.predict(x_test)
8 print('Training Accuracy :- {(model.score(variable) x_test: Any bund(2))}% \n')
9 print('Validation Accuracy :- {(model.score(x_test, y_test)*100).round(2)}% \n')

```

```

1
...
((769, 1500), (193, 1500), (769,), (193,))

```

```

1 #model training
2 model = OneVsRestClassifier(KNeighborsClassifier())
3 model.fit(x_train, y_train)

```

```

...
OneVsRestClassifier(estimator=KNeighborsClassifier())

```

Figure 22 : Code of Model Selection

2.2.4 Blogs

EMD Labs continuously improves our expertise in producing technical blogs that are relevant to our projects and areas of interest. In order to do that, I published more than four blogs once every two weeks about AI and ML, including

- Noise removal Techniques
- Support of AWS for NLP
- AWS Sage Maker
- Supervised ML

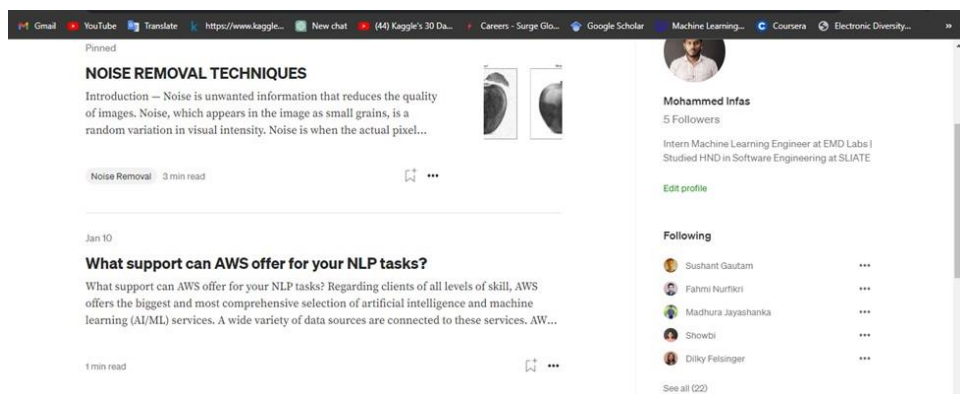


Figure 23 : Writing Blogs

CHAPTER 03

3.0 Introduction of internship training

3.1 Skill gained through the internship

3.1.1 Reading relevant technical literature

Technical literature relevant to the field was read to stay up to date with the latest trends and technologies. It should be remembered that improving practical skills is a continuous process, and it requires dedication and effort on one's part. Learning and practicing should be kept up, and improvement will be seen over time.

3.1.2 Participating in training and development programs

To acquire new skills and keep up with emerging technology, participants in training and development programs are offered by EMD Laboratories and Dream Space Academy.

3.1.3 Learning for others

Within the organization, mentors or other team leaders were sought out, and their experiences were drawn upon. Leadership philosophies were examined, and lessons from successes and failures were drawn.

3.1.4 Effective communication was developed

Good communication skills were established in order to facilitate successful communication with the team, stakeholders, and other departments. Both verbal and written communication fell under this category.

3.2 Conclusion

I choose to work for EMD Labs during my industrial placement term largely because I truly enjoy working on the software development side, where I can learn more about various technologies. I choose the EMD Labs back-end AI and ML division since I also enjoy participating in AI and ML development. This Branch is mainly focused on producing code as well as developing AI and ML on the back end. Also, the majority of the business was quite proud to be Aussie. After my first two to three weeks as an undergraduate trainee at the company, I attended technical training classes and was given an overview of the company's departments and process management in addition to our unique learning material. I'll simply discuss my experience here that I already went into length about in the previous chapter.

I got started by setting up my workspace before being given a project. I provided direction and small goals like finishing reading suggested resources by a certain day while I was mostly left alone during the studies. At EMD Labs, where they want to offer one job vacancy for AI and ML, I was the only individual receiving training in Python-based back-end AI and machine learning development. I rapidly became familiar with the back-end standards before being given the task of creating ML models to train various datasets. With this design, I was free to learn how to develop it and to become acquainted with back-end language.

I started out learning about Python-based back-end AI and ML programming. At first, I encountered a number of difficulties while training models and Python library version problems. My abilities to handle these mistakes and difficulties were afterwards developed. After some serious self-study and effort, I have begun working alone. For creative inspiration, I read a number of study papers and publications.

Most importantly, throughout my internship, I gained invaluable experience working with incredibly talented people who helped me achieve lofty objectives that, at times, I felt I would have failed at on my own. To unwind and have fun as a group, the training team scheduled movie days, parties, and snacks for us.

The training program greatly aided in the development of my soft skills, interpersonal and communication abilities, as well as my confidence in my ability to

succeed as a professional in the IT business. In conclusion, I would say that my time working as an undergraduate trainee at EMD Laboratories, residing with other very brilliant individuals, and gaining experience in the field we will be in the future was a tremendous time of my ATI life.

3.3 References

- (2022, 08 05). Retrieved from <https://vitalflux.com>: <https://vitalflux.com/resume-screening-using-machine-learning-nlp/>
- (2022, 07 10). Retrieved from <https://www.sciencedirect.com>:
<https://www.sciencedirect.com/science/article/pii/S187705092030750X>
- (2022, 07 10). Retrieved from <https://link.springer.com>:
https://link.springer.com/chapter/10.1007/978-981-33-4859-2_21
- (2022, 09 10). Retrieved from <https://ieeexplore.ieee.org>:
<https://ieeexplore.ieee.org/abstract/document/8473291>
- (2022, 10 07). Retrieved from <https://www.mccormick.northwestern.edu>:
<https://www.mccormick.northwestern.edu/electrical-computer/academics/graduate/masters/electrical-engineering/computer-vision-and-image-processing.html#:~:text=Computer%20vision%20uses%20image%20processing,the%20human%20visual%20cognitive%20system.>