



CSE3119 - ROBOTIC PROCESS AUTOMATION

Project Report

EFFICIENT EMAIL MANAGEMENT WITH UIPATH AUTOMATION SOLUTIONS

By

20BRS1270 ADAVELLI ROHAN REDDY

20BRS1218 PENMETS A GAJENDRA VARMA

20BRS1080 ARUN PRASATH B

Submitted to

Dr.Nithya Darisini

School of Computer Science and Engineering

April 2023

DECLARATION BY THE STUDENT

We, Adavelli Rohan Reddy, Penmetsa Gajendra Varma and Arun Prasath B bearing Reg. No. 20BRS1270, 20BRS1218 and 20BRS1080 hereby declare that this project report entitled EFFICIENT EMAIL MANAGEMENT WITH UIPATH AUTOMATION SOLUTIONS has been prepared by us towards the partial fulfilment of the requirement for the award of the Bachelor of Technology (B. Tech) Degree under the guidance of Dr. Nithya Darisini

We also declare that this project report is our original work and has not been previously submitted for the award of any Degree, Diploma, Fellowship, or other similar titles.

Place: Chennai

Date: 15th April 2023

ADAVELLI ROHAN REDDY (20BRS1270)

PENMETSA GAJENDRA VARMA (20BRS1218)

ARUN PRASATH B (20BRS1080)

ABSTRACT

The automated email processing system developed using UiPath and Python is a revolutionary technology that promises to transform the way emails are managed and processed. The system's advanced algorithms and machine learning capabilities enable it to identify patterns, extract key information, and classify emails based on their content and context. This in turn, enables the system to prioritize important emails, route them to the appropriate departments, and summarize them for easy review. Moreover, the system's spam detection module provides an added layer of security by identifying and flagging potentially malicious emails. With its ability to streamline email processing, reduce manual effort, and improve efficiency, the system is poised to become a game-changer in the realm of email management.

ACKNOWLEDGEMENT

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this project. It is not possible to prepare a project report without the assistance and encouragement of other people. This one is certainly no exception. On the very outset of this report, we would like to extend our sincere & heartfelt obligation towards all the personages who have helped us in this endeavor. Without their active guidance, help, cooperation & encouragement, we would not have made headway in the project. We are ineffably indebted to our faculty Dr. Nithya Darisini for her conscientious guidance and encouragement to accomplish this assignment presently. We extend our gratitude to VIT Chennai for giving us this opportunity. We also acknowledge with a deep sense of reverence, our gratitude towards our parents and members of our family, who have always supported us morally as well as economically.

At last, but not least gratitude goes to all of our friends who directly or indirectly helped us to complete this project report. Any omission in this brief acknowledgement does not mean lack of gratitude.

CONTENTS

1	Introduction	1
1.1	Objective and goal of the project	
1.2	Problem Statement.	
1.3	Motivation	
2	Literature Survey	1
3	Methodology	3
4	Implementation of System	4
5	Results & Discussion	17
6	Conclusion & Future Work	19
7	References	20

1. Introduction:

1.1 Objective and goal of the project:

The main objective of the project is to improve the efficiency and productivity of the email management system in a large organization. The project goal is to automate email processing and classification, thereby reducing the workload of employees and streamlining the email management process.

1.2 Problem Statement :

Email management is a time-consuming and tedious task in large organizations. The volume of emails received daily is overwhelming, making it difficult for employees to manage and respond to them promptly. The lack of a streamlined email management system leads to delays in communication, missed opportunities, and a decrease in overall productivity.

1.3 Motivation :

The motivation behind the project is to simplify the email management process for employees, reduce their workload, and improve the overall efficiency of the organization. Automation of email processing and classification can significantly reduce the time and effort required to manage emails, leading to a more streamlined workflow and increased productivity.

2. Literature Survey :

[1]Journal of critical reviews email client automation with RPA - November 2021 The businesses industry and other industries can simply automate the email processing task of segregating common emails into specific folders as per the organization's requirements. The proposed system in this paper is a smart email

client prototype that automates the task of email segregation and routing and by providing fixed reply using Robotic process automation (RPA) that runs on UiPath software [2] **Web Automation using Selenium Web driver Python - March 2019** Using automated testing tools like Selenium, GUI testing and Cross Browser testing and Web automation is done more effectively In this paper they performed Web Automation testing using Selenium web driver Python The web based application is tested with Selenium web driver with Python code. Cross Browser testing is done with the various leading browsers to check performance of applications as expected. [3] **Automation for Ease of Life : To Schedule Bulk Messaging and Email System using Python - November 2022** Machine mimics the human task, and reduces time. Humans can make mistakes in doing the same task repeatedly but machines cannot. Automation improves efficiency of work. In this Paper they have proposed Automation in sending Mails and Text Messages using Python.

3. Methodology :

- ❖ **Reading unread messages:** The "Get IMAP Mail Messages" activity is used to read unread messages. A for each loop is then used to read the body of each email and store it in a text file.
- ❖ **Summarization using Python:** The input text file is loaded and its contents are read. Any HTML tags present in the text are removed using the BeautifulSoup library. The text is then tokenized into sentences using the NLTK library, followed by filtering of unwanted sentences using regular expressions. Tokenization of the filtered sentences into words using NLTK is then performed. Stop words are removed from the text using NLTK's set of English stop words. Stemming of the words using NLTK's PorterStemmer algorithm is then carried out. A frequency distribution of the stemmed words

is created using NLTK's FreqDist class. The weighted frequency of each sentence is then calculated by summing the frequency of each word in the sentence. The average sentence score is calculated by dividing the total weighted frequency of all sentences by the number of sentences. A graph of the sentences based on cosine similarity is then built using the numpy library. The importance of each sentence is then calculated using the PageRank algorithm implemented in the network library. The top 3 ranked sentences are chosen as the summary, which is then written to another file as a single paragraph.

- ❖ **Writing summarized data to Excel:** The summarized text file is read and appended to a datatable using the "Add Data Row" activity. The datatable is then copied into Excel using the "Read Range" activity, which consists of sender email, subject, and summarized body information.
- ❖ **Department classification using Python:** The for loop checks for query words like "inquire" or "request" in the mail body. If present, the body is written to a text file and the department classifier python code is run. The input data is loaded from an Excel file using pandas library. A TF-IDF vectorizer is then created to transform the text data into numerical feature vectors. The target variable is created as the department column from the loaded Excel data. A Multinomial Naive Bayes classifier is then created and trained on the feature vectors and target variable. The trained classifier is saved to a file using pickle. A function named "predict_department" is defined to predict the department of an input text file. The function loads the trained classifier from the saved file using pickle. It reads the input text file and transforms it into a feature vector using the same TF-IDF vectorizer object.

Then, the department is predicted using the trained classifier and the predicted department is written to the output text file.

- ❖ **Forwarding email:** The output file is read, and the mail is forwarded accordingly to the respective department using the "Send SMTP Mail Message" activity.
- ❖ **Spam detection and organization:** Two activities are run parallelly using the "Parallel" activity. The first activity checks for words like "discount," "sale," "buy now," "limited offer," "lottery," "money-back," "risk-free," and "satisfaction guaranteed" in the mail body using regex. If found, the mail is stored in the Spams folder. In the second activity, it is checked whether the mail is sent from within the organization. If so, the mail is stored in the organization folder, and any attachments present in the mail are saved in the specified folder.

4. Implementation of System :

Code for Summarization Using Transformers:

```
import nltk

from nltk.corpus import stopwords

from nltk.tokenize import sent_tokenize

from nltk.tokenize import word_tokenize

from nltk.probability import FreqDist

from nltk.stem import PorterStemmer

from nltk.tokenize import word_tokenize
```

```

from nltk.cluster.util import cosine_distance

import numpy as np

import networkx as nx

from bs4 import BeautifulSoup

import re


# Load the input text file and read its contents

with open('C:/Users/Adavelli Rohan Reddy/Desktop/VIT/6th
sem/RPA/Project/Email_Auto_modern/Email/input.txt', 'r', encoding='utf-8-
sig') as file:

    text = file.read()


# Remove HTML tags from the text using BeautifulSoup

soup = BeautifulSoup(text, 'html.parser')

text = soup.get_text()


# Tokenize the text into sentences

sentences = sent_tokenize(text)


# Define patterns to filter out unwanted sentences

patterns = [r'^https?://', r'^www\.', r'^\w+@\w+\.\w+']

```

```
# Filter out unwanted sentences

filtered_sentences = []

for sent in sentences:

    if not any(re.match(pattern, sent) for pattern in patterns):

        filtered_sentences.append(sent)


# Tokenize the filtered sentences into words

words = word_tokenize(' '.join(filtered_sentences))


# Remove stop words from the text

stop_words = set(stopwords.words('english'))

filtered_words = [word for word in words if word.casefold() not in
stop_words]


# Perform stemming on the words

ps = PorterStemmer()

stemmed_words = [ps.stem(word) for word in filtered_words]


# Create a frequency distribution of the stemmed words
```

```

word_frequencies = FreqDist(stemmed_words)

# Calculate the weighted frequency of each sentence
weighted_frequencies = {}

for i, sent in enumerate(filtered_sentences):
    for word in word_tokenize(sent.lower()):
        if ps.stem(word) in word_frequencies:
            if i in weighted_frequencies:
                weighted_frequencies[i] += word_frequencies[ps.stem(word)]
            else:
                weighted_frequencies[i] = word_frequencies[ps.stem(word)]

# Calculate the average sentence score
total_sentences = len(weighted_frequencies)
average_score = sum(weighted_frequencies.values()) / total_sentences

# Build a graph of the sentences based on cosine similarity
sentence_graph = np.zeros((total_sentences, total_sentences))

for i in range(total_sentences):
    for j in range(total_sentences):

```

```

if i != j:

    sentence_graph[i][j] = (1 - cosine_distance(

        np.array(list(weighted_frequencies.values()))[i],

        np.array(list(weighted_frequencies.values()))[j]))

# Use PageRank to calculate the importance of each sentence

sentence_ranks = nx.pagerank(nx.from_numpy_array(sentence_graph))

# Choose the top 3 ranked sentences as the summary

summary_length = 3

top_sentences = sorted(((sentence_ranks[i], s) for i, s in
enumerate(filtered_sentences)), reverse=True)[:summary_length]

summary = ' '.join([s[1] for s in top_sentences])

# Write the summary to another file as a single paragraph

with open("C:/Users/Adavelli Rohan Reddy/Desktop/VIT/6th
sem/RPA/Project/Email_Auto_modern/Email/output.txt", 'w') as file:

    file.write(summary.replace('\n', ' '))

```

Code for Department Classification using Naïve Bayes:

```

import pandas as pd

```

```
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB

import pickle


# load the data

df = pd.read_excel("C:/Users/Adavelli Rohan Reddy/Desktop/VIT/6th
sem/RPA/Project/Email_Auto_modern/Email/Departments/Depts.xlsx",
engine="openpyxl")


# create a TF-IDF vectorizer object

tfidf = TfidfVectorizer(max_features=10000)


# transform the text data into numerical feature vectors

X = tfidf.fit_transform(df['Query'])


# create the target variable

y = df['Department']


# create a Naive Bayes classifier object

clf = MultinomialNB()
```

```

# fit the classifier to the data

clf.fit(X, y)


# save the model to a .sav file using pickle

with open("model.sav", "wb") as f:

    pickle.dump(clf, f)


# create a function to predict the department

"""def predict_department(text):

    with open("model.sav", "rb") as f:

        clf = pickle.load(f)

        X_test = tfidf.transform([text])

        return clf.predict(X_test)[0]"""


input = "C:/Users/Adavelli Rohan Reddy/Desktop/VIT/6th
sem/RPA/Project/Email_Auto_modern/Email/Departments/inp.txt"

output = "C:/Users/Adavelli Rohan Reddy/Desktop/VIT/6th
sem/RPA/Project/Email_Auto_modern/Email/Departments/out.txt"


def predict_department(input_file, output_file):

    with open("model.sav", "rb") as f:

```

```

clf = pickle.load(f)

# read the input text file

with open(input_file, 'r') as f:

    input_text = f.read()

X_test = tfidf.transform([input_text])

# predict the department

predicted_department = clf.predict(X_test)[0]

# write the predicted department to the output file

with open(output_file, 'w') as f:

    f.write(predicted_department)

predict_department(input,output)

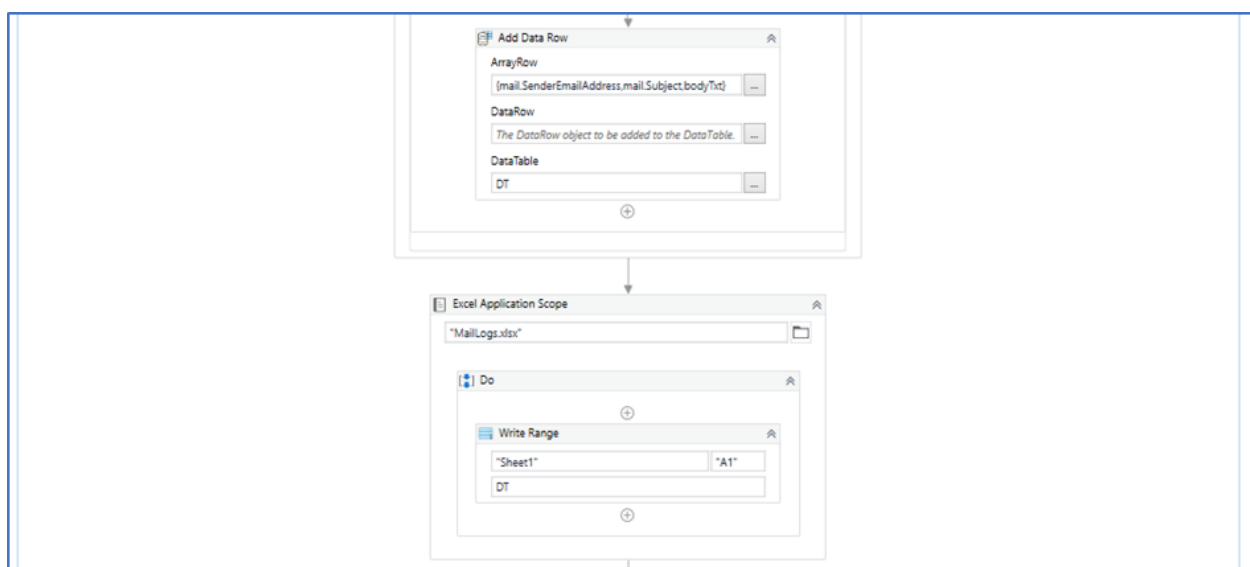
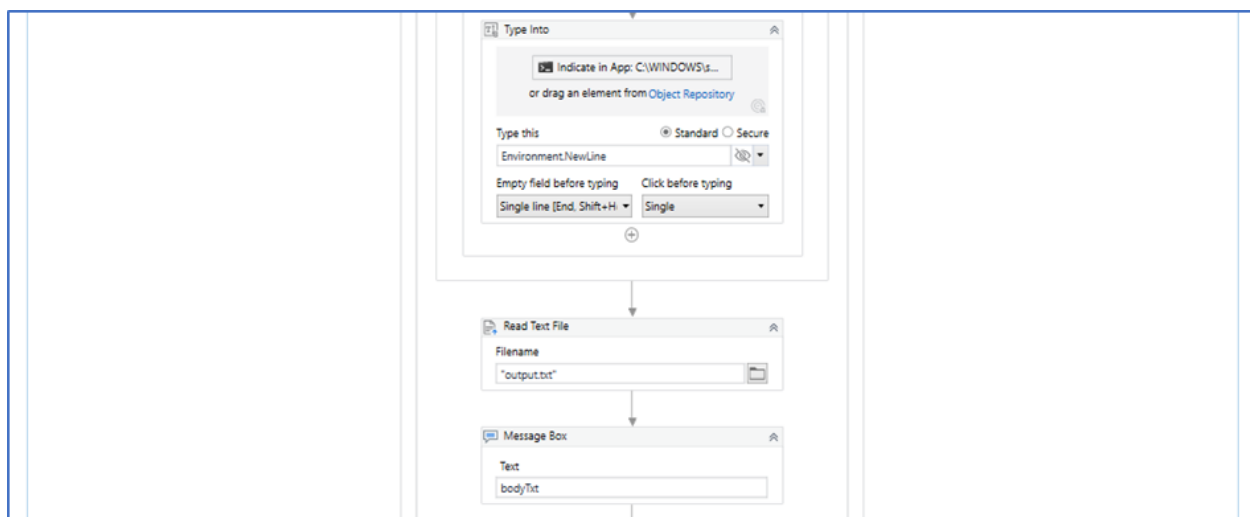
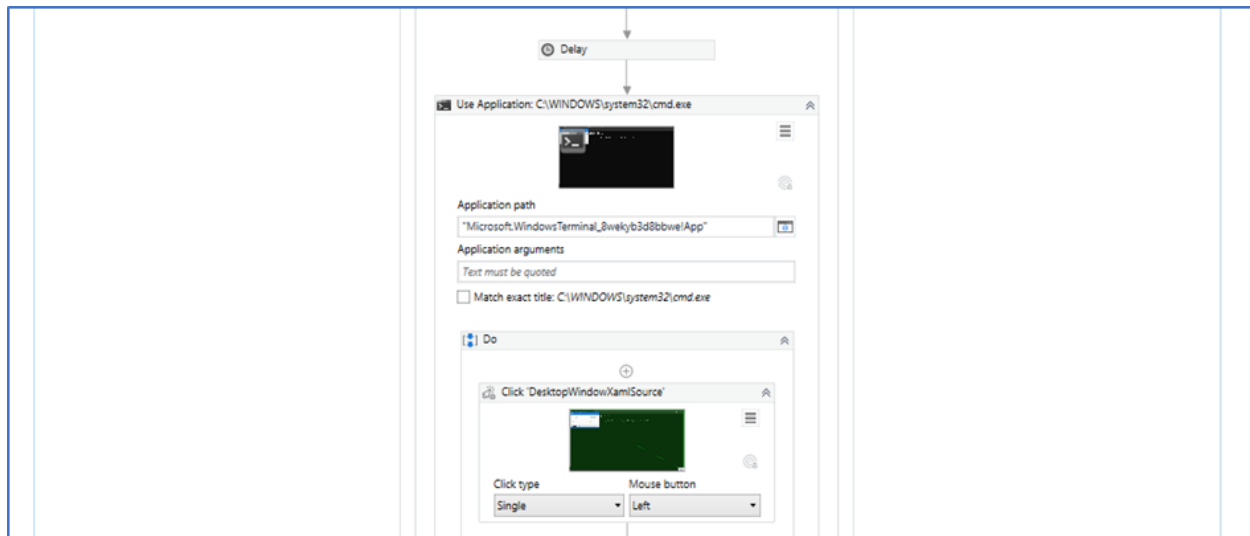
```

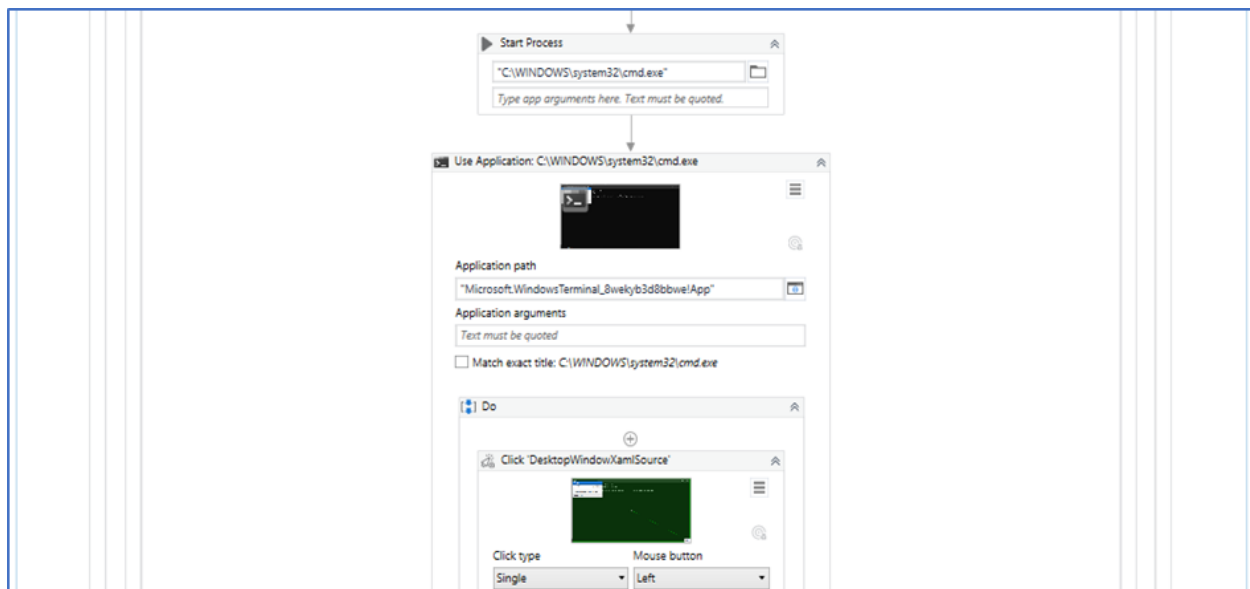
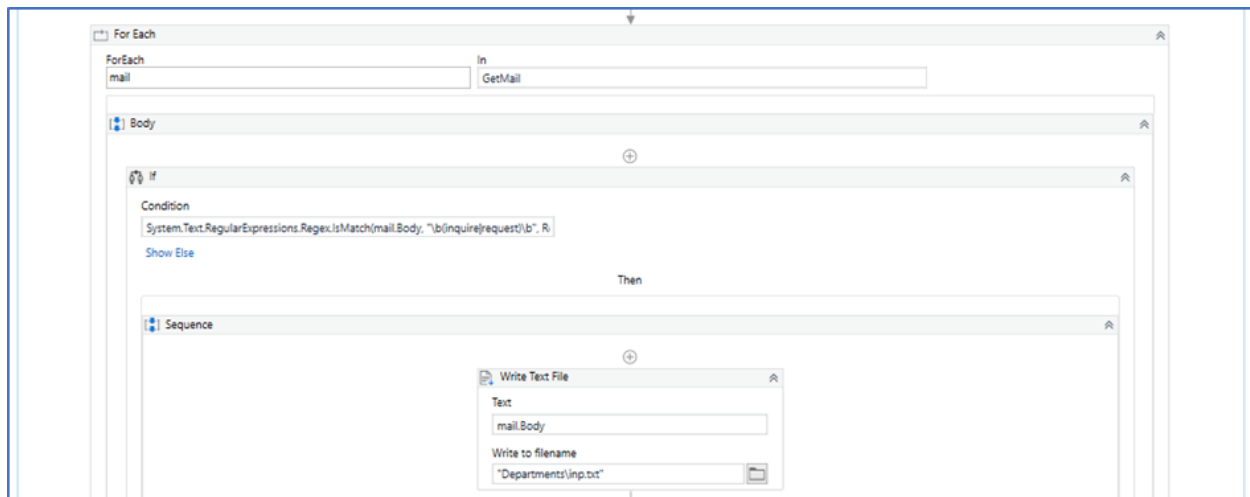
Departments Excel (Dataset used to Train Naïve bayes) :

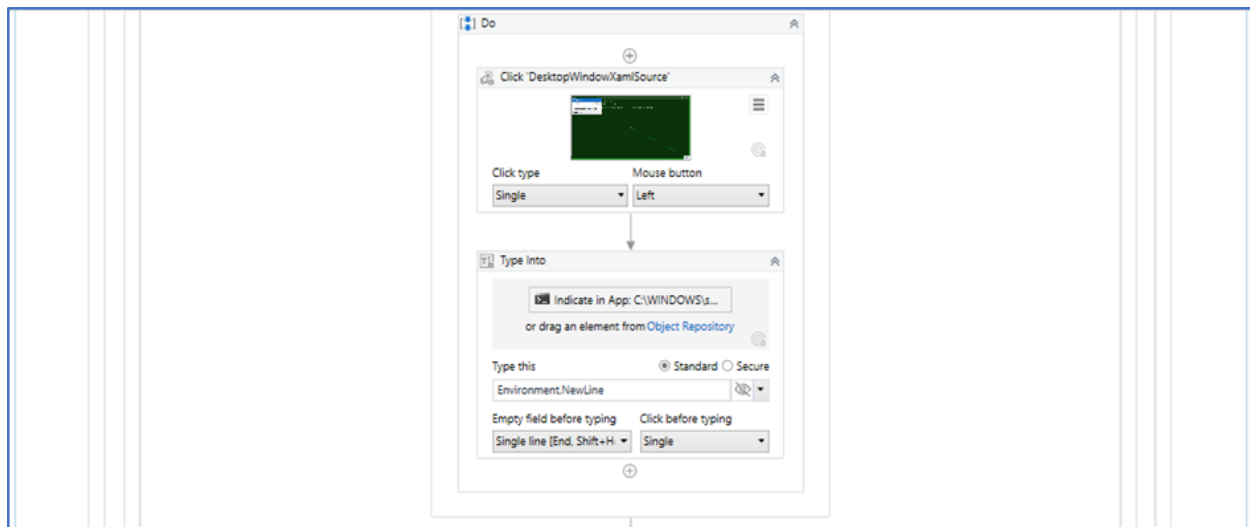
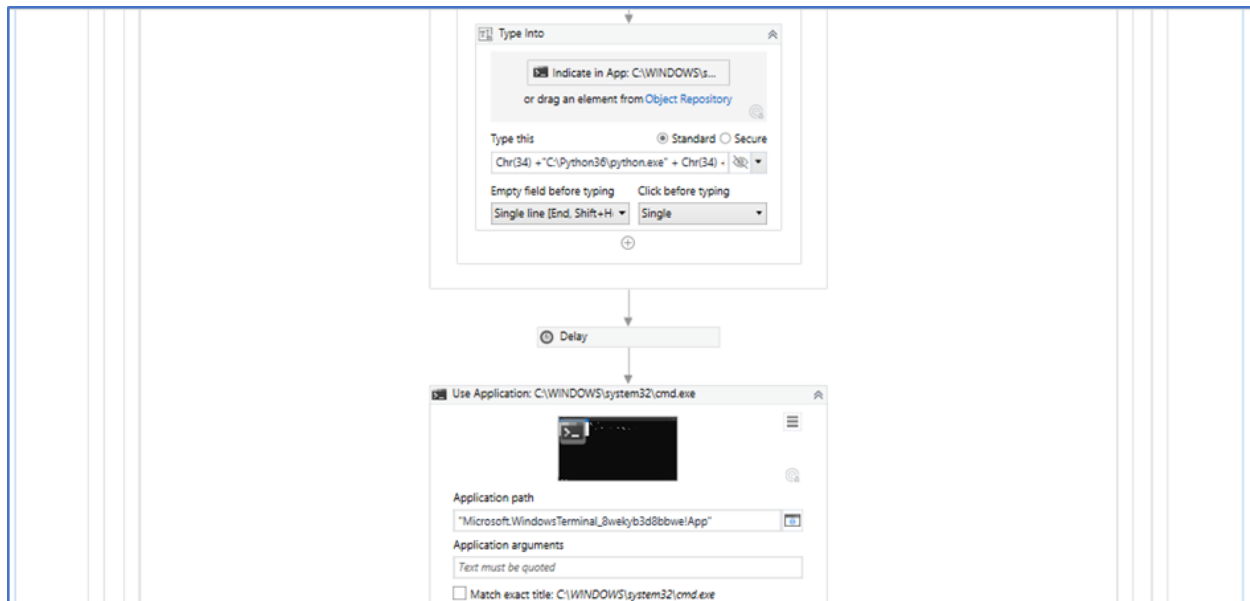
	A	B	C	D	E	F	G	H
1	Department	Query						
2	Admission:	What are the admission requirements for international students?						
3	Admission:	How do I apply for scholarships?						
4	Admission:	When is the deadline to apply for the upcoming semester?						
5	Financial A	What are the different types of financial aid available?						
6	Financial A	How do I fill out the FAFSA form?						
7	Registrar	How do I request a transcript?						
8	Registrar	What is the process for dropping a course?						
9	Student Se	Where can I find information about campus events?						
10	Student Se	How do I get involved in student clubs and organizations?						
11	Career Ser	How do I write a resume and cover letter?						
12	Career Ser	What job opportunities are available for my major?						
13	Academic	How do I schedule an appointment with my academic advisor?						
14	Academic	What courses do I need to take to fulfill my graduation requirements?						

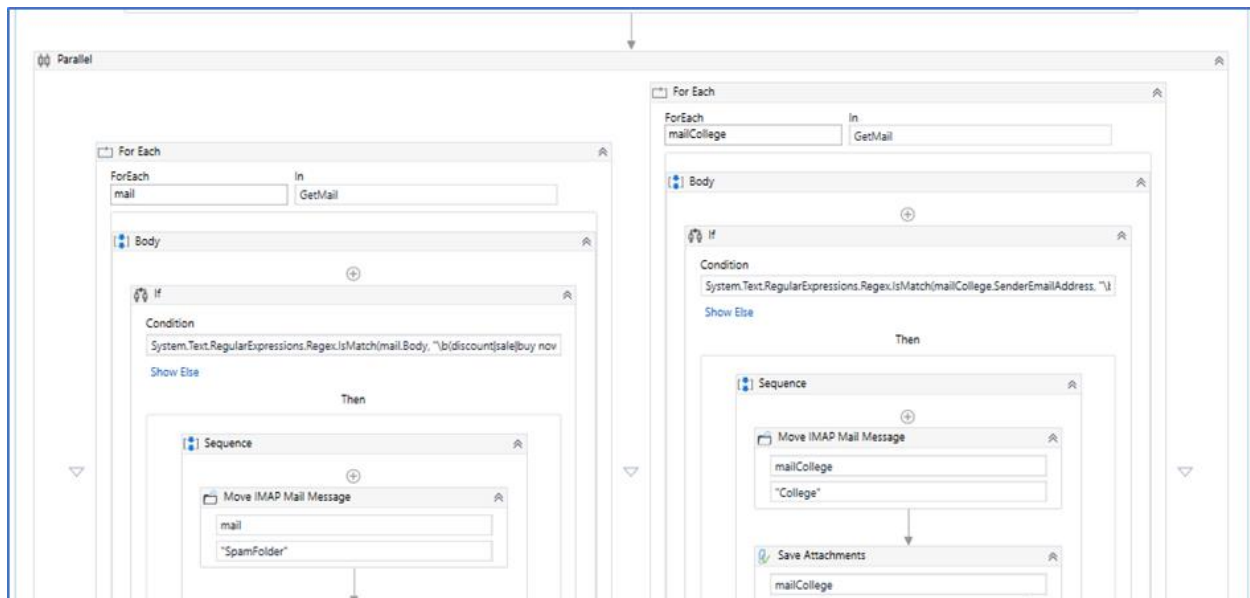
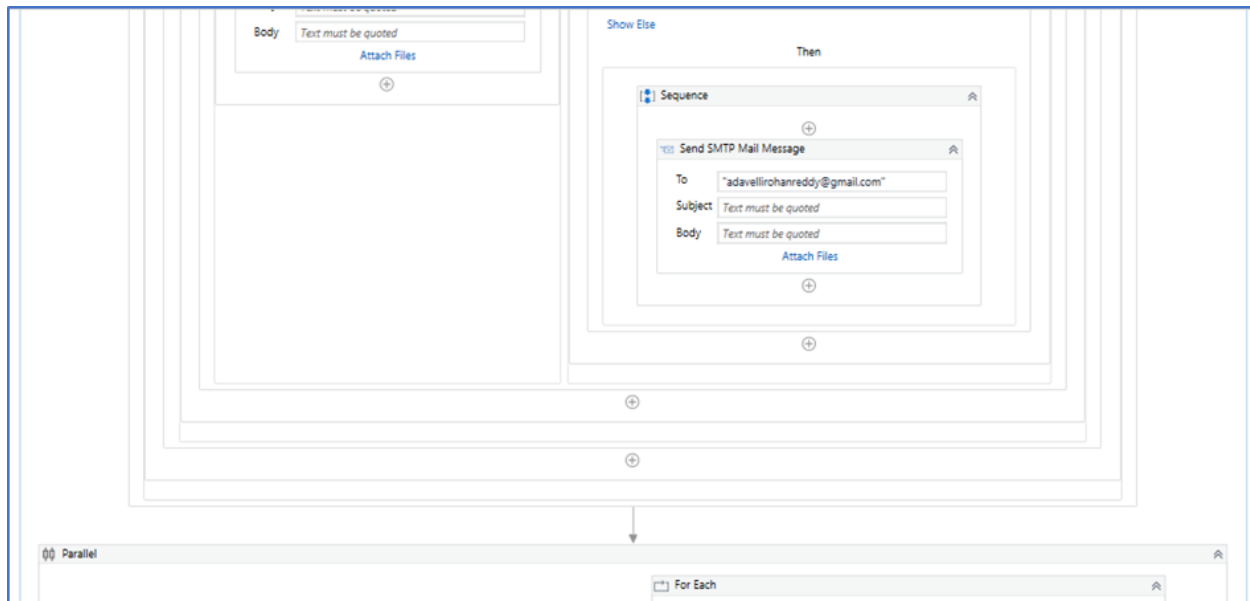
Workflow:











5. Results and Discussion :

Summarized Excel:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Sender	Subject	Body																			
2	arohan124	Promotion	We're only offering this discount for two days, so make sure to take advantage of it while you can. We're also offering a money-back guarantee on all purchases during this sale. This offer is only available to our email s																			
3	adavellirol	Subject: In We encourage everyone to participate and make this initiative a success. The contents of this email may contain legally protected confidential or privileged information of ♦Vellore Institute of Technology♦. In additio																				
4	arohan124	Housing	In dormitories, apartments, off-campus housing)Availability of on-campus housing for freshmen or transfer studentsRoommate matching processHousing costs and payment optionsMeal plan options and costsAny other r																			
5	arohan124	Admission	Sincerely,[Your Name] I look forward to hearing from you soon. I have researched [College Name] extensively and am impressed by the achievements and reputation of its faculty and students.																			
6	arohan124	Report	Models in time series are employed to predict the so2 readings in nearing years or months. In this paper, machine learning techniques are used to predict the concentration of so2 in the environment. With this increasin																			
7	no-reply@	2-Step Ver	♦ 2023 Google LLC. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA Iimage: GoogleI 2-Step Verification turned on tonvrpa0@gmail.com Your Google Account tonvrpa0@gmail.com is now protected with																			

Forwarded Mail to Concern Department:

Forwarded to Housing department mail based on the Text Classification using Naive bayes classifier

Fwd: Housing Inquiry in VIT CC



Tony Stark <tonyrpa0@gmail.com>

06:50



To: ARohan1245@gmail.com

----- Forwarded message -----

From: **A Rohan Reddy** <arohan1245@gmail.com>

Date: Sat, Apr 8, 2023 at 7:10 PM

Subject: Housing Inquiry in VIT CC

To: tonyrpa0@gmail.com <tonyrpa0@gmail.com>

Dear Housing Office,

I am writing to inquire about housing options available for students at [College Name] for the upcoming academic year. As I am planning to attend [College Name], I am interested in exploring the various housing options that the institution has to offer.

Could you please provide me with more information on the following:

Types of housing available (e.g. dormitories, apartments, off-campus housing)

Availability of on-campus housing for freshmen or transfer students

Fwd: Admissions Inquiry for VIT CC



Tony Stark <tonyrpa0@gmail.com>

06:50



To: adavellirohanreddy@gmail.com

----- Forwarded message -----

From: A Rohan Reddy <arohan1245@gmail.com>

Date: Sat, Apr 8, 2023 at 7:09 PM

Subject: Admissions Inquiry for VIT CC

To: tonyrpa0@gmail.com <tonyrpa0@gmail.com>

Dear Admissions Office,

I am writing to express my interest in applying to [College Name] for the upcoming academic year. I am excited about the possibility of studying at your institution and would like to request more information about the application process.

I am particularly interested in the [major/program] offered by [College Name] and believe that the curriculum and resources provided by your institution would allow me to achieve my academic and professional goals. I have researched [College Name] extensively and am impressed by the achievements and reputation of its faculty and students.

Could you please provide me with more information on the following:

We used UiPath activities and did the project. The project involves multiple activities to read, summarize, classify, and organize emails. The "Get IMAP Mail Messages" activity is used to read unread messages and a for each loop is used to read the body of each email and store it in a text file. The text is then summarized using Python by removing HTML tags, tokenizing into sentences, filtering unwanted sentences, removing stop words, stemming, and calculating the importance of each sentence using the PageRank algorithm. The top 3 ranked sentences are chosen as the summary and written to another file. The summarized text file is then written to

an Excel file using the "Read Range" activity. The department classification process involves checking for specific words in the email body, writing the body to a text file, and using the pandas library to load data from an Excel file. A Multinomial Naive Bayes classifier is trained on the feature vectors and target variable, and the trained classifier is saved to a file using pickle. A function is defined to predict the department of an input text file. The email is then forwarded to the respective department using the "Send SMTP Mail Message" activity. Spam detection is performed by checking for specific words in the email body, and any attachments present in the email are saved in a specified folder.

6. Conclusion and Future Work :

In conclusion, the developed automated email processing system using UiPath and Python has proven to be an effective tool for improving email management and reducing manual effort. The system's ability to read, summarize, and classify emails based on their content and context, as well as its spam detection and organization capabilities, have demonstrated its potential to revolutionize email processing in organizations. Furthermore, the integration of Python's machine learning capabilities has enabled the system to provide accurate and efficient results in email summarization and department classification.

However, there is still room for future work to enhance the system's capabilities. One area of improvement could be the implementation of a sentiment analysis module to classify emails based on their emotional tone. This could be particularly useful for customer service departments, enabling them to prioritize and respond to emails based on their level of urgency and emotional state. Additionally, the system could be further optimized for handling multilingual emails, which could help

organizations expand their global reach and improve communication with international clients.

7. REFERENCES :

1. Journal of critical reviews email client automation with RPA - November 2021
2. Web Automation using Selenium Web driver Python - March 2019
3. Automation for Ease of Life : To Schedule Bulk Messaging and Email System using Python - November 2022