

PHONE BANKING PHASE 2 - PROCESS DEFINITION DOCUMENT

Business Unit:	Phone Banking
Process Name:	Phone Banking Phase 2
Document Type:	Process Definition Document
Version No:	V1.1
Authors:	Arunav Sahay
Date Created:	12 August 2022
Last Amended:	27 September 2022
Project Phase:	Design
Prepared By:	ICICI – Wipro RPA Team

Version – 1

Introduction

Phone Banking team daily receives hundreds of emails from different customers querying about different products, issues, and services. The phone banking team has to manually look into every email and forward it to the desired team or take appropriate action. It is a tedious task to extract entities from the emails and look for what the customer is querying about. An RPA-AI based solution was henceforth designed to reduce the manual efforts and automate the entire process. An RPA was designed to fetch emails one by one and apply various business logic to identify the email components, product, nature, and entity of the email. This entire automation process was developed in Phase 1, where the bot was able to fetch the customer email and predict the product and intent of the email. Phase 2 of Phone Banking project was developed to predict sub-intent of the product and intent and extract major entities. There are around 88 sub-intents mapped in phase 2 of the project and 6 entities. The development was done to create an AI model which can classify the sub-intent of the email and extract the correct entity.

Procedure: EMAIL CLASSIFICATION

1. Access to Data

- a. The Business Team procures data from Genesys Team in the form of CSVs/Excels. It comprises of the Thread ID, Subject, Email Body, the tagged category/sub-category or product.
- b. Once the business team procures enough data, it is shared with the software team via different means like email, uploading the dataset on UIPATH Cloud or SFTP server.

2. Data Download and sampling

- a. Once the data is downloaded, the contents of the data samples are verified. Any unwanted excel or sheet is removed. The columns are looked at to see if they are properly labeled and have the 3 most important columns i.e., the email subject, body and label (product/intent/sub-intent).
- b. Same sub-intent data are clubbed together to form one single dataset.
- c. The columns of all the data files are inspected and changed if they are different. The 3 main columns should be **“Subject”**, **“Body”** & **“Sub-Category”**.

3. Data Separation & Bundling

- a. With the help of *pandas* library, we separate the different Sub-Category datasets into different excel/csv files. So, if there are 57 sub-intents to be trained there should be 57 dataset files, all having the 3 main columns **“Subject”**, **“Body”** & **“Sub-Category”**. These files are named after their sub-intents.

```
Charges_related-CC_credit_card_charges_-_others.xlsx
Charges_related-CC_enquiry_of_annual_joining_fee_for_credit_card.xlsx
Charges_related-CC_enquiry_on_autodebit_charges_for_credit_card.xlsx
Charges_related-CC_enquiry_on_credit_card_interest_charges.xlsx
Charges_related-CC_enquiry_on_fuel_surcharge_charges_for_credit_card.xlsx
Charges_related-CC_enquiry_on_late_payment_charges_for_credit_card.xlsx
Charges_related-CC_reversals_processed_lpc_int_overlimit_and_other_charges.xlsx
```

- b. Since all sub-intents have their intents or categories, each sub-intent data file is bundled under one folder of their intent.

Account Modification	account_modification_change_of_communication_address.csv
	account_modification_dormant_account_activation.csv
	account_modification_inactive_account_activation.csv
Charges_related-CC	Charges_related-CC_credit_card_charges_-_others.xlsx
	Charges_related-CC_enquiry_of_annual_joining_fee_for_credit_card.xlsx
	Charges_related-CC_enquiry_on_autodebit_charges_for_credit_card.xlsx
	Charges_related-CC_enquiry_on_credit_card_interest_charges.xlsx
	Charges_related-CC_enquiry_on_fuel_surcharge_charges_for_credit_card.xlsx
	Charges_related-CC_enquiry_on_late_payment_charges_for_credit_card.xlsx
	Charges_related-CC_reversals_processed_lpc_int_overlimit_and_other_charges.xlsx
Charges_related-RL	Charges_related-RL_cash_transaction_charges_savings_account.xlsx
	Charges_related-RL_debit_card_annual_fee.xlsx
	Charges_related-RL_ecs_return_charges_reason_for_ecs_return.xlsx
	Charges_related-RL_mab_savings_account_general_inquiry.xlsx
	Charges_related-RL_other_saving_account_charges.xlsx
	Charges_related-RL_reversal_of_debit_card_annual_fee.xlsx
	Charges_related-RL_reversal_of_ecs_return_charges.xlsx
	Charges_related-RL_savings_account_atm_pos_decline_charges.xlsx
Credit_Card_Payment_status-CC	Credit_Card_Payment_status-CC_auto_debit_status_.xlsx
	Credit_Card_Payment_status-CC_auto_debit_status.xlsx
	Credit_Card_Payment_status-CC_payment_modes_of_credit_card_tat_.xlsx
	Credit_Card_Payment_status-CC_payment_modes_of_credit_card_tat.xlsx
	Credit_Card_Payment_status-CC_payment_not_credited_.xlsx
	Credit_Card_Payment_status-CC_payment_not_credited.xlsx

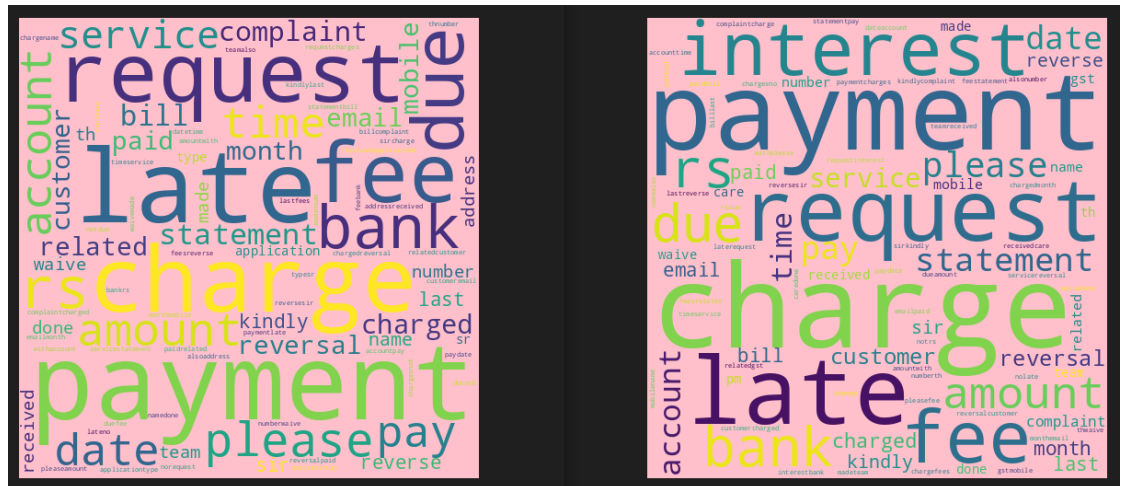
4. Exploratory Data Analysis (EDA)

- Once the data has been marked properly and bundled, we start the EDA. The major libraries/algorithms used in this process are defined in the following steps.
- Basic Data Cleaning is done before any EDA. Non-ascii characters, numbers, symbols, email addresses, URLs, certain words mentioned below are removed from the email sample. We also remove the words mentioned in the NLTK library.

```
from nltk.corpus import stopwords

en_stops_nltk = list(stopwords.words('english'))
```

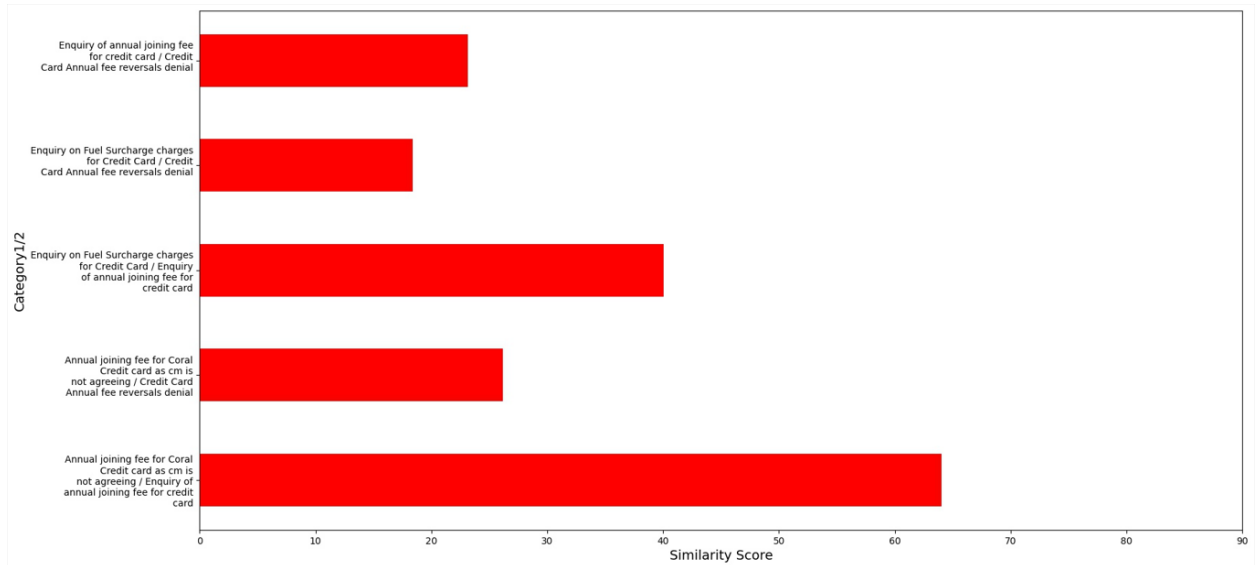
'a', 'about', 'above', 'after', 'again', 'against', 'all', 'am', 'amp', 'an', 'and', 'any', 'are', 'aren', 'as', 'at', 'be', 'because', 'been', 'before', 'being', 'below', 'between', 'both', 'but', 'by', 'can', 'click', 'complaint', 'd', 'dear', 'did', 'do', 'does', 'doing', 'don', 'down', 'during', 'each', 'feedback', 'few', 'for', 'forward', 'forwarded', 'from', 'further', 'gst', 'gt', 'had', 'has', 'have', 'having', 'he', 'hello', 'her', 'here', 'hers', 'herself', 'hi', 'him', 'himself', 'his', 'how', 'i', 'icici', 'if', 'in', 'inr', 'inr.', 'into', 'is', 'it', 'it's', 'its', 'itself', 'just', 'level1complaint', 'll', 'lt', 'm', 'ma', 'maam', 'madam', 'me', 'more', 'most', 'my', 'myself', 'nbsp', 'now', 'o', 'of', 'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours', 'ourselves', 'out', 'over', 'own', 'please', 'pls', 'quot', 're', 'regards', 'rs', 'rs,', 'rs.', 'rs..', 'rupees', 's', 'same', 'she', 'she's', 'should', 'sincerely', 'sir', 'so', 'some', 'such', 't', 'th', 'than', 'thank', 'thanks', 'that', 'that'll', 'the', 'their', 'theirs', 'them', 'themselves', 'then', 'there', 'these', 'they', 'this', 'those', 'through', 'to', 'too', 'under', 'until', 'up', 've', 'very', 'was', 'we', 'were', 'what', 'when', 'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'wrote', 'x', 'x.', 'xd', 'xx', 'y', 'you', 'you'd', 'you'll', 'you're', 'you've', 'your', 'yours', 'yourself', 'yourselves'



- e. Another plot is generated to understand the similarities between each subcategory. A cosine similarity is calculated between the TFIDF Vectors of the set of the data word.

```
def generate_similarity_scores(word_list_dict):
    for index, subcat1 in enumerate(word_list_dict.keys()):
        subcat1_list = word_list_dict[subcat1]
        subcat1_word = ' '.join(subcat1_list)
        for subcat2 in list(word_list_dict.keys())[index+1:]:
            #print(index, " ", subcat1, " ", subcat2)
            subcat2_list = word_list_dict[subcat2]
            subcat2_word = ' '.join(subcat2_list)
            #word_set = set(subcat1_list).union(subcat2_list)
            word_set = [subcat1_word, subcat2_word]
            vectorizer = TfidfVectorizer()
            final_features = vectorizer.fit_transform(word_set)
            #print(final_features[0,:].shape, final_features[1,:].shape)
            feature1 = final_features.getrow(0)
            feature2 = final_features.getrow(1).transpose()
            similarity = (feature1*feature2)[0,0]
            print(index, " ", subcat1, " ", subcat2, " ", similarity)
            #print(similarity)
            print('-----')
```

The similarity scores between the 2 subcategories are plotted in the following manner:



- f. All word clouds are studied, and an inference is drawn on how similar the dataset is and what model would fit the best to train the data.

5. Training Dataset Generation

- It was found from various experiments that 900 sample count was enough for training a good model. In this step we'll generate a single training file where each sub-category will have a maximum of 900 data samples and the rest will be saved in the validation file.
- All the files mentioned in point 3.b. are read in a for loop and merged into a bigger dataframe. Duplicates are removed and after giving a random shuffle, the training dataset is generated. With the samples with count under or equal to 900 saved as training dataframe and the rest saved as validation dataframe.
- The *Subject* and *Body* column are merged to form the *text* column.
- A new column is created label which has the labels of the sub-intents. The sub-intents are renamed from another file, which has the format of the label in the following way:

New Subcategory Name = CategoryName_SubCategoryName

Subcategory	Subcategory Labels	Category
Charges_related-CC_credit_card_charges_-_others_	Charges Related - CC_Credit Card Charges - Others	Charges Related - CC
Charges_related-CC_enquiry_of_annual_joining_fee_for_credit_card_	Charges Related - CC_Enquiry Of Annual Joining Fee For Credit Card	Charges Related - CC
Charges_related-CC_enquiry_on_autodebit_charges_for_credit_card_	Charges Related - CC_Enquiry On Autodebit Charges For Credit Card	Charges Related - CC
Charges_related-CC_enquiry_on_credit_card_interest_charges_	Charges Related - CC_Enquiry On Credit Card Interest Charges	Charges Related - CC
Charges_related-CC_enquiry_on_fuel_surcharge_charges_for_credit_card_	Charges Related - CC_Enquiry On Fuel Surcharge Charges For Credit Card	Charges Related - CC
Charges_related-CC_enquiry_on_late_payment_charges_for_credit_card_	Charges Related - CC_Enquiry On Late Payment Charges For Credit Card	Charges Related - CC
Charges_related-CC_reversals_processed_lpc_int_overlimit_and_other_charges_	Charges Related - CC_Reversals Processed LPC/Int Overlimit And Other Charges	Charges Related - CC

Here the Subcategory is the old subcategory Name, and the Subcategory Labels column is the new name.

- Finally, the new file consists of only 2 columns, namely **text** and **label**.

text	label
Liabilities Imobile 000WBaGNB2EHUA0U:Level1Complaint / Mobile banking Name: Shalini Manda Ganesh Account No / Application No: 086601518465 Product/Service: Mobile banking Request related to: iMobile Type of request/complaint: iMobile related issues E-mail address: shaliniganesh07@gmail.com Mobile no:9677873793 Telephone no: +91-96-77873793 Level 2 Service Request(SR) No: Complaint/Feedback: Unable to open the imobile app message displayed in the screen we are currently experiencing difficulty in processing your request please try again after some time	Imobile_Imobile Technical Issue SR
Liabilities Imobile 000WBaGNB2EHU8DK:Gold line don't renew bu imobile sap External Email Warning: Do not click on any attachment or links/URL in this email unless sender is reliable. Dear sir I am not renewal my gold lone by imobile application my registra mobile number is 7073257766 and my lone number is 379305000436 please chek	Imobile_Imobile Technical Issue SR
Liabilities Imobile 000WBaGNB2EHU0HK: External Email Warning: Do not click on any attachment or links/URL in this email unless sender is reliable. Account No:024301531408 Mobile No:8125913079 Can I know How to create upi ID in imobile app for icici bank	Imobile_Imobile Technical Issue SR

6. Training the model – Development Environment

- The dataset file (csv) is copied in the dataset folder.
- The Python Environments is activated with the libraries mentioned in the SDD document.
- The train.py file is called using the python command *python train.py*. It starts the training of the model and after some time the model is generated and saved in the *model* folder.
- A training and testing report is generated which is viewed to understand the training accuracy and performance with the test data.

7. Validating the model – Development Environment

- The trained model is validated by calling the main.py file and passing the data from the validation file generated in point 5.a. The function is called in the following way:

```
from main import Main
mn = Main()
ec_res = mn.predict(text)
```

- The variable ec_res stores the predicted class and the confidence value.
- The predicted class is then compared to the actual label of the data and if the value is correct, then the correct counter of that label is increased by 1 else if the value is incorrect, then the incorrect counter is increased by 1.
- The predicted confidence is also stored in a list for correct and incorrect predictions and then the average is taken out.
- Finally, the total number of matched and unmatched percentages are calculated and presented in the form of a report.

	Sub-Category	Matched	Unmatched	Total Samples	Avg Matched Confidence	Avg Unmatched Confidence	Matched_Perc	Unmatched_Perc
1								
2	Statement - CC_Credit Card E-Statement Not Received Request	3155	107	3262	94.55	81.81	96.72	3.28
3	Charges Related - CC_Reversals Processed LPC/Int Overlimit And Other Charges	1995	545	2540	77.95	65.67	78.54	21.46
4	Charges Related - RL_Debit Card Annual Fee	996	86	1082	80.88	56.23	92.05	7.95
5	Charges Related - CC_Enquiry On Late Payment Charges For Credit Card	2480	396	2876	85.29	65.88	86.23	13.77
6	Imobile_Imobile Technical Issue SR	886	1213	2099	69.19	71.01	42.21	57.79
7	Transaction Status_Transaction Status Declined And Status-CC	1073	594	1667	76.31	66.59	64.37	35.63
8	Charges Related - CC_Enquiry On Credit Card Interest Charges	1415	96	1511	93.11	60.55	93.65	6.35
9	Charges Related - CC_Enquiry Of Annual Joining Fee For Credit Card	409	17	426	98.51	60.29	96.01	3.99
10	Charges Related - RL_Mab Savings Account General Inquiry	1512	232	1744	88.59	59.05	86.7	13.3
11	Charges Related - RL_Other Saving Account Charges	714	225	939	68.83	59.29	76.04	23.96
12	Imobile_Imobile Activation Issue-Probe Handset Details	1191	117	1308	86.36	65.66	91.06	8.94
13	Credit Card Payment Status_Payment Not Credited	2809	132	2941	91.73	67.89	95.51	4.49
14	Debit Card - Cheque Deliverables_Debit Card Cheque Book Not Received	586	1707	2293	62.7	70.3	25.56	74.44
15	UPI/IMPS/NEFT/RTGS_Upi Imps-Chargeback SR	2414	497	2911	79.59	62.92	82.93	17.07
16	Charges Related - RL_ECS Return Charges Reason For ESC Return	289	92	381	74.53	62.79	75.85	24.15
17	Account Modification_Dormant Account Activation	358	16	374	93	68	95.72	4.28
18	Account Modification_Change Of Communication Address	229	3	232	98.13	67.33	98.71	1.29
19	Account Closure_Account closure procedure	146	146	292	65.32	67.55	50	50
20	Internet Banking login - RL_Know USER ID/Activation of USER ID	877	1	878	99.39	25	99.89	0.11
21	Debit Card - Cheque Deliverables_Re Dispatch Of Debit Card Cheque Book	610	123	733	80.89	55.21	83.22	16.78

f. This validation report is then shared with the business team.

8. Initial testing and analysis

- After sharing the validation report to the business team, they share with us more testing data which is unlabeled.
- Again main.py file is called with input as string (Subject + Body) and the predicted label, and the score is saved as new columns in a new excel file.
- This report is again shared with the business team, which they return with the correct and incorrect accuracies and suggestions.
- The suggestions and changes are reviewed and either the preprocessing or the hyperparameters are tuned and re-trained. Sometimes, even new ML algorithms are researched and retrained.
- This cycle continues until the business team gets satisfied with the outcome of the model.

9. Sharing of codes and dataset file

- The code and the dataset files are shared to the deployment team via SFTP, cloud or email, to upload the same in the UIPATH AIFABRIC.

Procedure: ENTITY EXTRACTION

1. Access to Data

- The data used for Email Classification is used again for Entity Extraction.
- Amount, Date, Mobile Number, Deliverable Type and Type of payment are required to be extracted from the emails. We select those email sub-categories which has majority of these entities, such as *“Enquiry on Credit Card Interest Charges”*, *“Payment Not Credited”*, *“Cheque Book Dispatch Status”*, *“Wrong Fund Transfer”*
- The data from these files are collected and combined. We start by taking at least 200 email samples from each of the subcategories mentioned above.

2. Data Filtration and Preparation

- The *“Subject”* and *“Body”* columns are concatenated, and all other columns are deleted. We then create 5 new empty columns *Date*, *Amount*, *MobileNo*, *ModeOfPayment*, *DeliverableType* for entities annotations.
- We clean the data by removing the multiple spaces, asterisk symbol, single-double quotations, brackets, new line characters. The cleaned dataset is then saved in a CSV file with encoding as *“latin-1”* or *“utf-8”*.

3. Annotations

- The text which is considered for annotation is copied and pasted as elements of a list under that entity column. If there are no annotations for a particular entity, we put an empty list as its value.

Statements	Date	Amount	MobileNo	ModeOfPayment	DeliverableType
Name: Riyas Ahamed N Account No / Application No: 4748466858374004 Product/Service: Credit Card Request related to: Others Type of request/complaint: Other issues E-mail address: n.riyasnavas@gmail.com Mobile no:9941405191 Telephone no: +91-- Level 2 Service Request(SR) No: Complaint/Feedback: Dear Icici Team, I am writing this to you for consideration in waiving the charges made on my credit card(4748466858374004) for being late in paying the amount of Rs. 3,038.91. I am going to pay the total outstanding amount on or before my due date of this month. Please revise the extra charges(Rs.3038.91) on my credit card. I am ready to pay the amount tomorrow (30th July22). Please do the needful. I shall be ever grateful if I get an affirmative response for the waiving of bank charges on my credit card. Thanks. Regards, Riyas Ahamed N					
	[“30th July22”]	[“Rs. 3,038.91”]	[“9941405191”]		

- The case sensitivity must be maintained, and the columns names must be *“Statements”*, *“Date”*, *“Amount”*, *“MobileNo”*, *“ModeOfPayment”* & *“DeliverableType”*.
- The file is then saved as a CSV file.

4. Training the model – Development Environment

- The dataset file is copied into the *dataset* folder of the project directory.
- The python library environment is activated. The libraries used in this project have been mentioned in the SDD Document.
- Finally, we run this command *python train.py* to start the training. Since the algorithm used here is Spacy which is a Deep Neural Network algorithm, it takes a lot of time getting trained.
- Once the model is trained, it is saved in the *model/entity_extraction_model* folder.

5. Validating the model – Development Environment

- The test data is used to validate the model after calling the main.py file.
- The output from main.py file is saved in the excel report and the following columns are saved.

Date: Results are extracted from model (class date) then formats the date into standard date format i.e **dd-mm-yyyy**

DateRange: Results are extracted from model (class date) then formats the date into standard date format i.e **dd-mm-yyyy to dd-mm-yyyy**

Amount: Results are extracted from model (class Amount) + **find_entity.py** (After cleaning & regex patterns) then formats to only numbers & decimal (rs,inr,k,lacs,etc)

MobileNo: Results are extracted from model (class MobileNo) + added patterns(regex) in spacy then removing mobileno -> length of mobileno < 10 & > 12

ModeOfPayment: Results are extracted from model (class ModeOfPayment)

DeliverableType: Results are extracted from model (class DeliverableType)

SRNumber: Results are extracted by patterns(regex) in spacy

- c. The report file is observed and if needed changes are made in the dataset file by adding more annotations to increase the accuracy of the entity extraction.
- d. Once the entity extraction accuracy looks strong enough, we share the observations with the client.

6. Initial testing and analysis

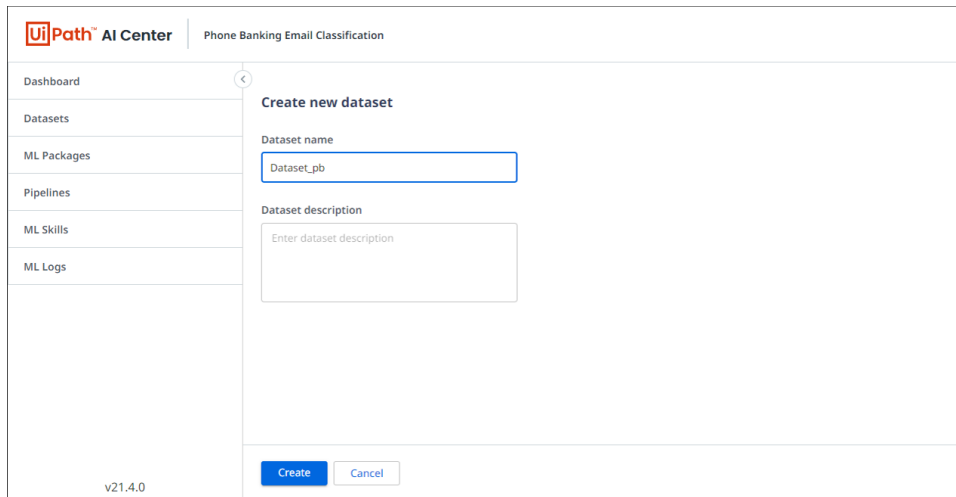
- a. After sharing the validation report to the business team, they share with us more testing data which is unlabeled.
- b. We execute the steps from 5.a to 5.b and save the report.
- c. This report is shared with the business team, which they return with the correct and incorrect accuracies and suggestions.
- d. The suggestions and changes are reviewed and either the preprocessing or the hyperparameters are tuned and re-trained. Sometimes, even new ML algorithms are researched and retrained.
- e. This cycle continues until the business team gets satisfied with the outcome of the model.

7. Sharing of codes and dataset file

- a. The code and the dataset files are shared to the deployment team via SFTP, cloud or email, to upload the same in the UIPATH AIFABRIC.

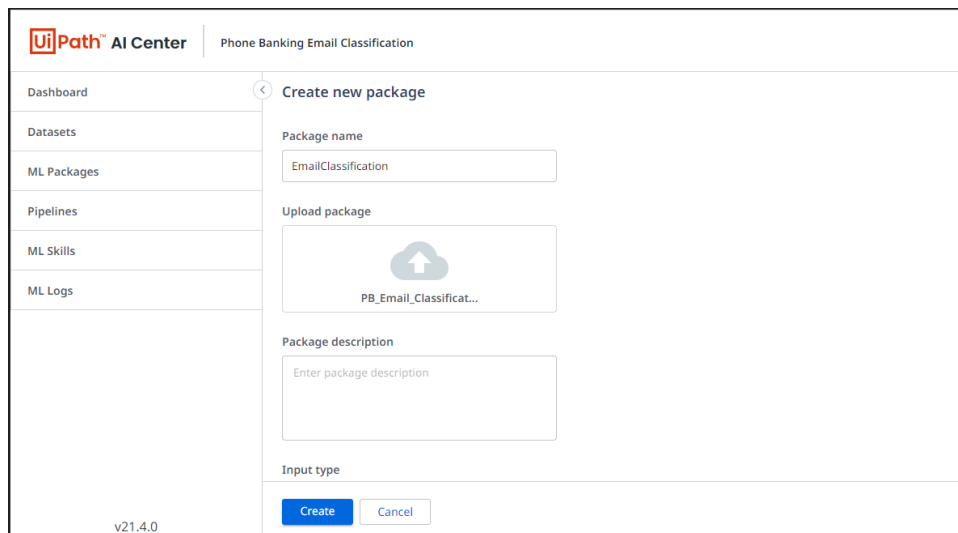
Procedure: UIPATH UPLOAD

1. In the following steps we will discuss the creation of ML SKILL in UIPATH AIFABRIC.
2. Open the URL: <https://172.28.138.149:31390/ai-app/>
3. Create a new project.
4. Create a new Dashboard and add the name and description details.



The screenshot shows the 'Create new dataset' form in the UiPath AI Center. The left sidebar contains a navigation menu with options: Dashboard, Datasets, ML Packages, Pipelines, ML Skills, and ML Logs. The main content area is titled 'Create new dataset' and includes a 'Dataset name' field with the value 'Dataset_pb', a 'Dataset description' field with the placeholder 'Enter dataset description', and 'Create' and 'Cancel' buttons at the bottom. The top header shows the UiPath AI Center logo and the page title 'Phone Banking Email Classification'. The version 'v21.4.0' is displayed in the bottom left corner.

5. Click on ML Packages option and enter the details. Upload the code in zip file in Upload Package option. Only the code and library files will get uploaded.



The screenshot shows the 'Create new package' form in the UiPath AI Center. The left sidebar contains a navigation menu with options: Dashboard, Datasets, ML Packages, Pipelines, ML Skills, and ML Logs. The main content area is titled 'Create new package' and includes a 'Package name' field with the value 'EmailClassification', an 'Upload package' section with a cloud upload icon and the file name 'PB_Email_Classificat...', a 'Package description' field with the placeholder 'Enter package description', and an 'Input type' field. The 'Create' and 'Cancel' buttons are at the bottom. The top header shows the UiPath AI Center logo and the page title 'Phone Banking Email Classification'. The version 'v21.4.0' is displayed in the bottom left corner.

6. Click on Pipelines and create a new pipeline. Select the package which you just uploaded under choose package option.

UiPath AI Center Phone Banking Email Classification

Dashboard < Create new pipeline run

Datasets

ML Packages

Pipelines

ML Skills

ML Logs

Pipeline type
Full Pipeline run

Choose package
PB_EmailClassification_0206

Choose package major version
4

Choose package minor version
1

Choose input dataset
PB_Email_dataset_36K x

Create Cancel

v21.4.0

7. Enter the parameters as mentioned in the screenshot.

UiPath AI Center Phone Banking Email Classification

Dashboard < Create new pipeline run

Datasets

ML Packages

Pipelines

ML Skills

ML Logs

Pipeline type
Full Pipeline run

Choose package
PB_EmailClassification_0206

Choose package major version
4

Choose package minor version
1

Choose input dataset
PB_Email_dataset_36K x

Create Cancel

v21.4.0

8. Finally add the ML Skill and choose the package name.

UiPath™ AI Center | Phone Banking Email Classification

Dashboard

Datasets

ML Packages

Pipelines

ML Skills

ML Logs

Create New ML Skill

Name

PB_email_MLSkill

Choose Package

PB_EmailClassification_0206

Choose package major version

4

Choose package minor version

1

Skill description

Enter skill description

Create Cancel

v21.4.0

9. Click Create. The Create New ML Skill page is closed, and the ML Skills page is displayed, containing the new skill with Deploying status.
10. If the deployment is successful, the status of the skill changes from Deploying to Available.
11. The API is created after successful deployment which is then used in the