```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
%cd /content/drive/MyDrive/Precog
```

```
/content/drive/MyDrive/Precog
```

```
!pip install pdf2image
```

```
Collecting pdf2image
  Downloading https://files.pythonhosted.org/packages/03/62/089030fd16ab3e5c245315d63c86
Requirement already satisfied: pillow in /usr/local/lib/python3.6/dist-packages (from pd
Installing collected packages: pdf2image
Successfully installed pdf2image-1.14.0
```

```
!pip install tabula
```

```
Collecting tabula
  Downloading https://files.pythonhosted.org/packages/eb/bf/b2084620900655a8c080af9873cc
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from tab
Building wheels for collected packages: tabula
  Building wheel for tabula (setup.py) ... done
  Created wheel for tabula: filename=tabula-1.0.5-cp36-none-any.whl size=10586 sha256=c2
  Stored in directory: /root/.cache/pip/wheels/47/2c/e2/33c0445cb41b20cf2dc01d31664d62ca
Successfully built tabula
Installing collected packages: tabula
Successfully installed tabula-1.0.5
```

```
!apt-get install poppler-utils
```

```
!sudo apt install tesseract-ocr
```

Task A

```
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv
try:
```

```
try:
    from PIL import Image
except ImportError:
    import Image
# import pytesseract
from google.colab.patches import cv2_imshow
import pdf2image

pdf = "./Rec_Task/1c1edeee-a13e-4b2e-90be-eb1dd03c3384.pdf"
# pdf = "./Rec_Task/a6b29367-f3b7-4fb1-a2d0-077477eac1d9-1.pdf"
# pdf = "./Rec_Task/d9f8e6d9-660b-4505-86f9-952e45ca6da0.pdf"
'''
Not working for below pdf
'''
# pdf = "./Rec_Task/EICHERMOT.pdf"

image = ""

images = pdf2image.convert_from_path(pdf)
# images
for image in images:
  image_path = pdf[:-4]+".jpg"

image_matrix = cv2.imread(image_path)
image_matrix = cv2.cvtColor(image_matrix, cv2.COLOR_BGR2GRAY)

height, width = image_matrix.shape

edges = cv2.adaptiveThreshold(image_matrix,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINA
cv2_imshow(edges)

edges = ~edges
edges = cv2.dilate(edges, np.ones((4,4)))
edges = cv2.morphologyEx(edges, cv2.MORPH_OPEN, np.ones((2,2)))
cv2_imshow(edges)


horizontal_lines = cv2.morphologyEx(edges, cv2.MORPH_OPEN, np.ones((1,width//20)))
vertical_lines = cv2.morphologyEx(edges, cv2.MORPH_OPEN, np.ones((height//20,1)))

combined = cv2.add(horizontal_lines, vertical_lines)
cv2_imshow(combined)

cell_points = cv2.bitwise_and(horizontal, vertical)
cell_points = cv2.dilate(cell_points, np.ones((4,4)))

cv2_imshow(cell_points)
```

April 05 , 2018

To, The Manager, Listing Department, BSE Limited, Phiroze Jeejeebhoy Tower, Dalal Street, Mumbai 400 001. BSE Scrip Code: 532636	To, The Manager, Listing Department, The National Stock Exchange of India Ltd., Exchange Plaza, 5 Floor, Plot C/1, G Block, Bandra - Kurla Complex, Bandra (E), Mumbai 400 051. Tel No.: 2659 8235 Fax No.: 26598237/ 26598238 NSE Symbol: IIFL

Dear Sir/ Madam,

Subject: Schedule of Analysts/Institutional Investor Meeting

Pursuant to the Regulation 30(6) of the SEBI (Listing Obligations and Disclosure Requirements) Regulations, 2015, we would like to inform you the following schedule of Analysts/Institutional Investor Meeting with the Company:

Date	Particulars	Type of Interaction
April 06, 2018	SBICAP Securities Ltd	One on One

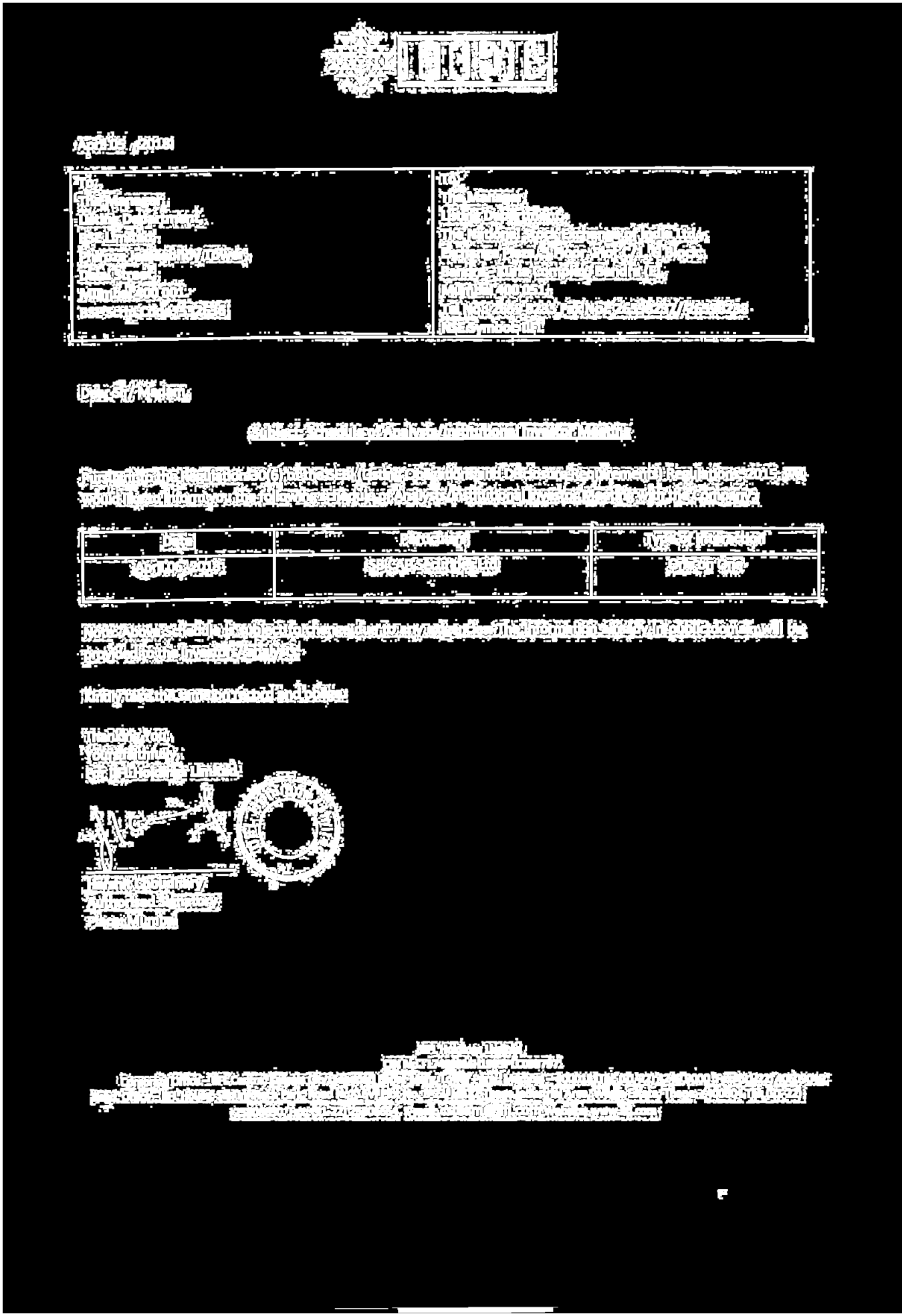
Note: Above Schedule is subject to change due to any exigencies. The information already in public domain will be provided to the investors/ analysts.

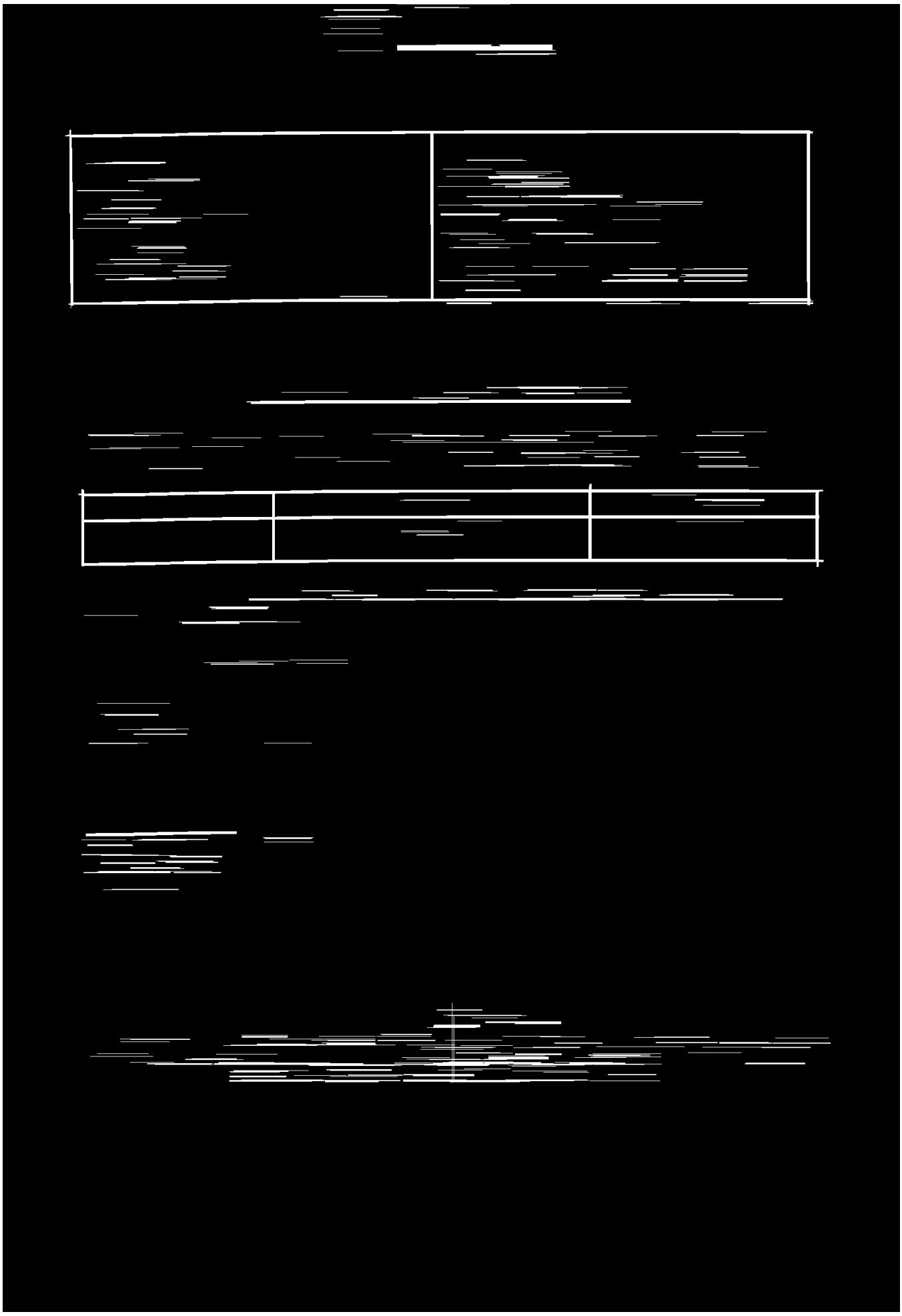
Kindly take the same on record and oblige.

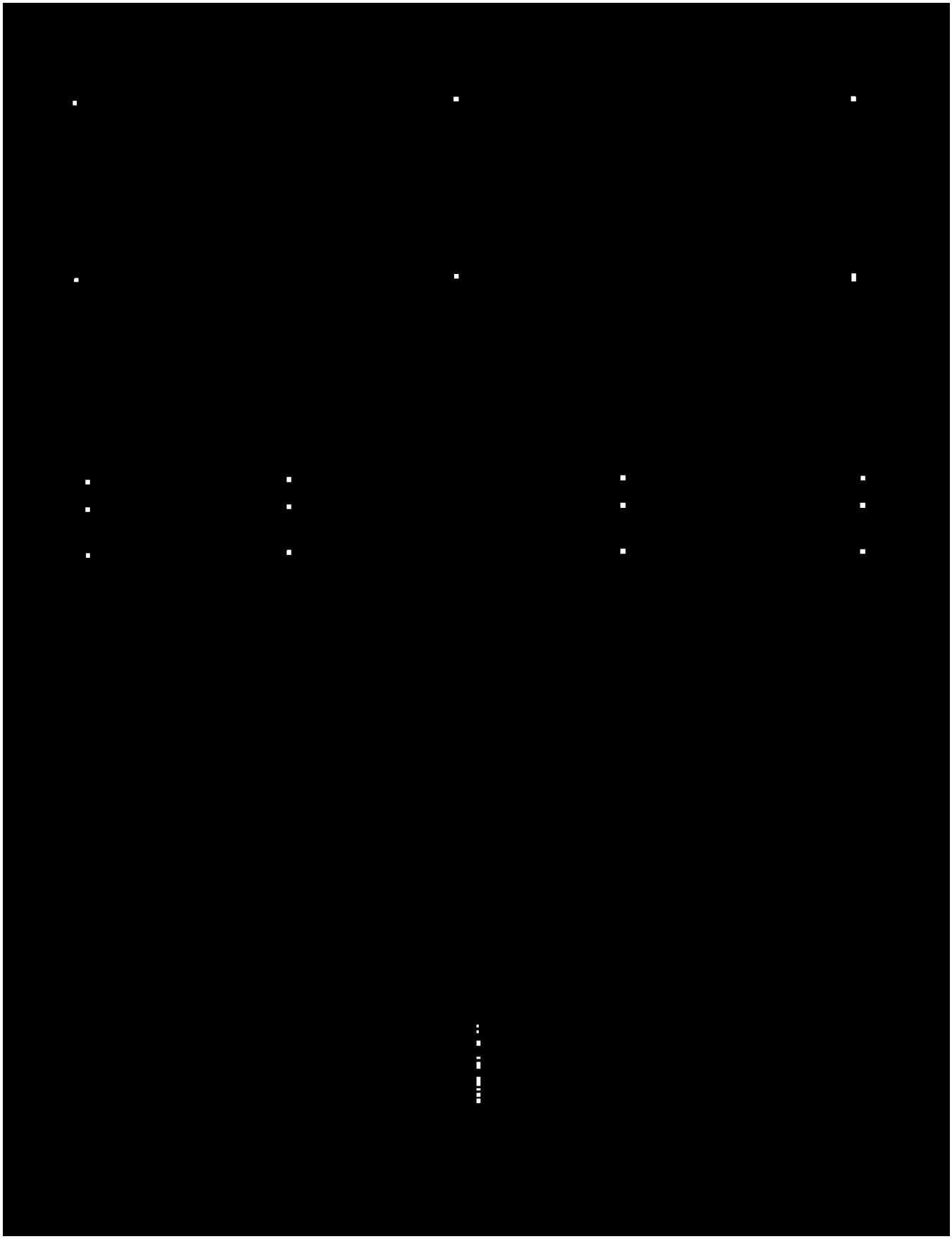
Thanking You,
Yours faithfully,
For IIFL Holdings Limited

Harshit Choudhary
Authorised Signatory
Place: Mumbai

IIFL Holdings Limited
CIN No.: L74999MH1995PLC093797
Corporate Office – IIFL Centre, Kamala City, Senapati Bapat Marg, Lower Parel, Mumbai – 400013. Tel: (91-22) 4249 9000. Fax: (91-22) 40609049
Regd. Office – IIFL House, Sun Infotech Park, Road No. 16V, Plot No. B-23, MIDC, Thane Industrial Area, Wagle Estate, Thane – 400604. Tel: (91-22) 25806650. Fax: (91-22) 25806654. E-mail: csteam@iifl.com Website: www.iifl.com







```
table_contours, hierarchy = cv2.findContours(combined, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SI
cell_contours, hierarchy = cv2.findContours(combined, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)


TableContours = []

for contour in table_contours:
  if cv2.contourArea(countour) >= 10000:
      TableContours.append(cv2.boundingRect(countour))

print("Number of tables in the pdf ", len(TableContours))


CellContours = []

for countour in cell_contours:
  if cv2.contourArea(countour) >= 5000:
      CellContours.append(cv2.boundingRect(countour))


     Number of tables in the pdf  2
```

Approach

The above code extracts the horizontal and vertical lines from a given pdf, saves them in different and then adds the 2 arrays (here images). This gives us the third image shown, which helps us visuaize tables in the image. Taking 'bitwise and' gives us the corners of the tables available in the pdf. That might be helpful afterwards.

'countours' used above is nothing but the bounding boxes of the boxes retrieved on taking 'bitwise or' of the images.

'tab_countours' used above is again the bounding boxes but it just takes into account the big boxes, as a table can have many bounding boxes within itself plus it can have a bounding box on a whole, so only the bounding box around the table will be saved in tab_countours.

Via a threshold on the contourArea, I got just the tables and the cells inside them. I have vertices for all of them.

Further, the cells can be mapped to tables with the help of their coordinates. This way we will have the cells present in a particular table. Now using pyteserract library of python the data within the

cells can be retrieved. And since we know the cells within a table, we can easily make a dictionary of the rows in that table and save it in the MongoDB.

Task B

```
'''
Populating db
'''
!python3 task3.py


'''
making wordcloud of top 10 tags
'''
!python3 find_tags.py
!python3 make_wordcloud_tags.py


'''
Draw a barplot for the tags
'''
!python3 plotting.py
```

Inferences

The above code :

- reads data from the XML files, converts into dictionary and then stores them in a collection in MongoDB.
- reads tags from the 'tags' collection and stores its frequency in a file.
- the tags with their respective frequency is read from the file and a wordcloud is made out of it.
- a bar graph is plotted to show the frequency of the top 10 most-frequent hashtags.

By a close analysis of the wordcloud and the tag frequency list, I came to the conclusion that the subsampling was done on the basis of questions relating to "Web development" which includes front-end, back-end as well as server-side engineering (DevOps).