Analytics Vidhya

Data Extraction from Unstructured PDFs

BEGINNER LIBRARIES PANDAS PYTHON UNSTRUCTURED DATA

This article was published as a part of the Data Science Blogathon

Introduction:

Data Extraction is the process of extracting data from various sources such as CSV files, web, PDF, etc. Although in some files, data can be extracted easily as in CSV, while in files like unstructured PDFs we have to perform additional tasks to extract data.

There are a couple of Python libraries using which you can extract data from PDFs. For example, you can use the *PyPDF2* library for extracting text from PDFs where text is in a sequential or formatted manner i.e. in lines or forms. You can also extract tables in PDFs through the *Camelot* library. In all these cases data is in structured form i.e. sequential, forms or tables.

However, in the real world, most of the data is not present in any of the forms & there is no order of data. It is present in unstructured form. In this case, it is not feasible to use the above python libraries since they will give ambiguous results. To analyze unstructured data, we need to convert it to a structured form.

As such, there is no specific technique or procedure for extracting data from unstructured PDFs since data is stored randomly & it depends on what type of data you want to extract from PDF.

Here, I will show you a most successful technique & a python library through which you can extract data from *bounding boxes* in unstructured PDFs and then performing data cleaning operation on extracted data and converting it to a structured form.

PyMuPDF:

I have used the PyMuPDF library for this purpose. This library provided many applications such as extracting images from PDF, extracting texts from different shapes, making annotations, draw a bounded box around the texts along with the features of libraries like *PyPDF2*.

Now, I will show you how I extracted data from the bounding boxes in a PDF with several pages.

Here are the PDF and the red bounding boxes from which we need to extract data.

Traffic Crash Report

*DENOTES MANDATORY FIELD FOR SUPPLEMENT REPORT

☒ PHOTOS TAKEN ☒ OH-2 □ OH-3
□ SECONDARY CRASH □ OH-1P □ OTHER
□ PRIVATE PROPERTY

LOCAL INFORMATION: 3F

LOCAL REPORT NUMBER *: 2100904 8

REPORTING AGENCY NAME *: MANSFIELD POLICE

NCIC *: 07001

HIT/SKIP: 1 - SOLVED 2 - UNSOLVED

NUMBER OF UNITS: 01

UNIT IN ERROR: 98 — 98 - ANIMAL 99 - UNKNOWN

COUNTY *: 70

LOCALITY *: 1 — 1. CITY 2. VILLAGE 3. TOWNSHIP

LOCATION: CITY, VILLAGE, TOWNSHIP *: MANSFIELD

CRASH DATE / TIME *: 04252021 0528

CRASH SEVERITY: 5
1 - FATAL
SERIOUS INJURY
2 - SUSPECTED
MINOR INJURY
3 - SUSPECTED
4 - INJURY POSSIBLE
5 - PROPERTY DAMAGE ONLY

ROUTE TYPE: | ROUTE NUMBER: | PREFIX: 1 (1 - NORTH 2 - SOUTH 3 - EAST 4 - WEST) | LOCATION ROAD NAME: MAIN | ROAD TYPE: ST

LATITUDE: 40.79 06 20

ROUTE TYPE: | ROUTE NUMBER: | PREFIX: (1 - NORTH 2 - SOUTH 3 - EAST 4 - WEST) | REFERENCE ROAD NAME (ROAD, MILEPOST, HOUSE #): 1150 | ROAD TYPE:

LONGITUDE: 82.51 32 10

REFERENCE POINT: 3 — 1 - INTERSECTION 2 - MILE POST 3 - HOUSE #

DIRECTION FROM REFERENCE: 1 - NORTH 2 - SOUTH 3 - EAST 4 - WEST

DISTANCE FROM REFERENCE:

DISTANCE UNIT OF MEASURE: 1 - MILES 2 - FEET 3 - YARDS

ROUTE TYPE	ROAD TYPE		
IR - INTERSTATE ROUTE (TP)	AL - ALLEY	HW - HIGHWAY	RD - ROAD
US - FEDERAL US ROUTE	AV - AVENUE	LA - LANE	SQ - SQUARE
SR - STATE ROUTE	BL - BOULEVARD	MP - MILEPOST	ST - STREET
CR - NUMBERED COUNTY ROUTE	CR - CIRCLE	OV - OVAL	TE - TERRACE
TR - NUMBERED TOWNSHIP ROUTE	CT - COURT	PK - PARKWAY	TL - TRAIL
	DR - DRIVE	PI - PIKE	WA - WAY
	HE - HEIGHTS	PL - PLACE	

INTERSECTION RELATED
□ WITHIN INTERSECTION OR ON APPROACH
□ WITHIN INTERCHANGE AREA
NUMBER OF APPROACHES:

ROADWAY
□ ROADWAY DIVIDED

LOCATION OF FIRST HARMFUL EVENT: 1
1 - ON ROADWAY
2 - ON SHOULDER
3 - IN MEDIAN
4 - ON ROADSIDE
5 - ON GORE
6 - OUTSIDE TRAFFIC WAY
7 - ON RAMP
8 - OFF RAMP
9 - CROSSOVER
10 - DRIVEWAY/ALLEY ACCESS
11 - RAILWAY GRADE CROSSING
12 - SHARED USE PATHS OR TRAILS
13 - BIKE LANE
14 - TOLL BOOTH
99 - OTHER / UNKNOWN

MANNER OF CRASH COLLISION/IMPACT: 1
1 - NOT COLLISION BETWEEN TWO MOTOR VEHICLES IN TRANSPORT
2 - REAR-END
3 - HEAD-ON
4 - REAR-TO-REAR
5 - BACKING
6 - ANGLE
7 - SIDESWIPE, SAME DIRECTION
8 - SIDESWIPE, OPPOSITE DIRECTION
9 - OTHER / UNKNOWN

DIRECTION OF TRAVEL:
1 - NORTH
2 - SOUTH
3 - EAST
4 - WEST

MEDIAN TYPE:
1 - DIVIDED FLUSH MEDIAN (< 4 FEET)
2 - DIVIDED FLUSH MEDIAN (≥ 4 FEET)
3 - DIVIDED, DEPRESSED MEDIAN
4 - DIVIDED, RAISED MEDIAN (ANY TYPE)
9 - OTHER/UNKNOWN

□ WORK ZONE RELATED
□ WORKERS PRESENT
□ LAW ENFORCEMENT PRESENT
□ ACTIVE SCHOOL ZONE

WORK ZONE TYPE:
1 - LANE CLOSURE
2 - LANE SHIFT/CROSSOVER
3 - WORK ON SHOULDER OR MEDIAN
4 - INTERMITTENT or MOVING WORK
5 - OTHER

LOCATION OF CRASH IN WORK ZONE:
1 - BEFORE THE 1st WORK ZONE WARNING SIGN
2 - ADVANCE WARNING AREA
3 - TRANSITION AREA
4 - ACTIVITY AREA
5 - TERMINATION AREA

CONTOUR: 1
1 - STRAIGHT LEVEL
2 - STRAIGHT GRADE
3 - CURVE LEVEL
4 - CURVE GRADE
9 - OTHER/UNKNOWN

CONDITIONS: 1
1 - DRY
2 - WET
3 - SNOW
4 - ICE
5 - SAND, MUD, DIRT, OIL, GRAVEL
6 - WATER (STANDING, MOVING)
7 - SLUSH
9 - OTHER/UNKNOWN

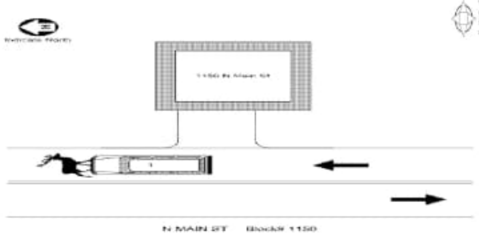
SURFACE: 2
1 - CONCRETE
2 - BLACKTOP, BITUMINOUS, ASPHALT
3 - BRICK/BLOCK
4 - SLAG, GRAVEL, STONE
5 - DIRT
9 - OTHER/UNKNOWN

LIGHT CONDITIONS: 3
1 - DAYLIGHT
2 - DAWN/DUSK
3 - DARK - LIGHTED ROADWAY
4 - DARK - ROADWAY NOT LIGHTED
5 - DARK - UNKNOWN ROADWAY LIGHTING
9 - OTHER / UNKNOWN

WEATHER: 2
1 - CLEAR
2 - CLOUDY
3 - FOG, SMOG, SMOKE
4 - RAIN
5 - SLEET, HAIL
6 - SNOW
7 - SEVERE CROSSWINDS
8 - BLOWING SAND, SOIL, DIRT, SNOW
9 - FREEZING RAIN OR FREEZING DRIZZLE
99 - OTHER/UNKNOWN

NARRATIVE

U#1 was Northbound on N Main St at 1150. A deer crossed the road Eastbound and U#1 struck it causing damage.



Write an "N" on the compass diagram to indicate the direction of north.

Drawing Not To Scale

CRASH REPORTED DATE/TIME	DISPATCH DATE/TIME	ARRIVAL TIME	SCENE CLEARED DATE / TIME
042520210528	042520210532	042520210536	042520210547

REPORT TAKEN BY
☒ POLICE AGENCY
□ MOTORIST

TOTAL TIME ROADWAY CLOSED	OTHER INVESTIGATION TIME	TOTAL MINUTES
		15

OFFICER'S NAME*: CAROLYN YOUNG

OFFICER'S BADGE NUMBER*: 1754

CHECKED BY OFFICER'S NAME*: P. WILLIAMS

CHECKED BY OFFICER'S BADGE NUMBER*: 1051

□ SUPPLEMENT (CORRECTION OR ADDITION TO AN EXISTING REPORT SENT TO ODPS)

MOTORIST / NON-MOTORIST

LOCAL REPORT NUMBER *
2100904 8

UNIT #: 1
NAME: LAST, FIRST, MIDDLE: STANTO, QUINN NICOLE
DATE OF BIRTH: 04131980
AGE: 41
GENDER: F

ADDRESS: CITY, STATE, ZIP: 1754 BROWNSTONE BLVD - H, TOLEDO, OH 43601
CONTACT PHONE - INCLUDE AREA CODE:

INJURIES: 5
INJURED TAKEN BY:
EMS AGENCY (NAME):
INJURED TAKEN TO: MEDICAL FACILITY (NAME, CITY):
SAFETY EQUIPMENT USED: 4
□ DOT-COMPLIANT MC HELMET
SEATING POSITION: 1
AIR BAG USAGE: 1
EJECTION: 1
TRAPPED: 1

OL STATE: OH
OPERATOR LICENSE NUMBER: TK825051
OFFENSE CHARGED:
LOCAL CODE:
OFFENSE DESCRIPTION:
CITATION NUMBER:

OL CLASS: 4
ENDORSEMENT SELECT UP TO 2:
RESTRICTION SELECT UP TO 3:
DRIVER DISTRACTED BY: 1
ALCOHOL/DRUG SUSPECTED: □ ALCOHOL □ MARIJUANA □ OTHER DRUG
CONDITION: 1
ALCOHOL TEST: STATUS: 1 TYPE: VALUE: .
DRUG TEST(S): STATUS: 1 TYPE: RESULTS SELECT UP TO 4:

UNIT #:
NAME: LAST, FIRST, MIDDLE:
DATE OF BIRTH:
AGE:
GENDER:

ADDRESS: CITY, STATE, ZIP:
CONTACT PHONE - INCLUDE AREA CODE:

INJURIES:
INJURED TAKEN BY:
EMS AGENCY (NAME):
INJURED TAKEN TO: MEDICAL FACILITY (NAME, CITY):
SAFETY EQUIPMENT USED:
□ DOT-COMPLIANT MC HELMET
SEATING POSITION:
AIR BAG USAGE:
EJECTION:
TRAPPED:

OL STATE:
OPERATOR LICENSE NUMBER:
OFFENSE CHARGED:
LOCAL CODE:
OFFENSE DESCRIPTION:
CITATION NUMBER:

OL CLASS:
ENDORSEMENT SELECT UP TO 2:
RESTRICTION SELECT UP TO 3:
DRIVER DISTRACTED BY:
ALCOHOL/DRUG SUSPECTED: □ ALCOHOL □ MARIJUANA □ OTHER DRUG
CONDITION:
ALCOHOL TEST: STATUS: TYPE: VALUE: .
DRUG TEST(S): STATUS: TYPE: RESULTS SELECT UP TO 4:

UNIT #:
NAME: LAST, FIRST, MIDDLE:
DATE OF BIRTH:
AGE:
GENDER:

ADDRESS: CITY, STATE, ZIP:
CONTACT PHONE - INCLUDE AREA CODE:

INJURIES:
INJURED TAKEN BY:
EMS AGENCY (NAME):
INJURED TAKEN TO: MEDICAL FACILITY (NAME, CITY):
SAFETY EQUIPMENT USED:
□ DOT-COMPLIANT MC HELMET
SEATING POSITION:
AIR BAG USAGE:
EJECTION:
TRAPPED:

OL STATE:
OPERATOR LICENSE NUMBER:
OFFENSE CHARGED:
LOCAL CODE:
OFFENSE DESCRIPTION:
CITATION NUMBER:

OL CLASS:
ENDORSEMENT SELECT UP TO 2:
RESTRICTION SELECT UP TO 3:
DRIVER DISTRACTED BY:
ALCOHOL/DRUG SUSPECTED: □ ALCOHOL □ MARIJUANA □ OTHER DRUG
CONDITION:
ALCOHOL TEST: STATUS: TYPE: VALUE: .
DRUG TEST(S): STATUS: TYPE: RESULTS SELECT UP TO 4:

INJURIES

1 - FATAL
2 - SUSPECTED SERIOUS INJURY
3 - SUSPECTED MINOR INJURY
4 - POSSIBLE INJURY
5 - NO APPARENT INJURY

INJURED TAKEN BY

1 - NOT TRANSPORTED / TREATED AT SCENE
2 - EMS
3 - POLICE
9 - OTHER / UNKNOWN

SAFETY EQUIPMENT

1 - NONE USED
2 - SHOULDER BELT ONLY USED
3 - LAP BELT ONLY USED
4 - SHOULDER & LAP BELT USED
5 - CHILD RESTRAINT SYSTEM - FORWARD FACING
6 - CHILD RESTRAINT SYSTEM - REAR FACING
7 - BOOSTER SEAT
8 - HELMET USED
9 - PROTECTIVE PADS USED (ELBOW, KNEES, ETC.)
10 - REFLECTIVE CLOTHING
11 - LIGHTING - PEDESTRIAN / BICYCLE ONLY
99 - OTHER / UNKNOWN

SEATING POSITION

1 - FRONT - LEFT SIDE (MOTORCYCLE DRIVER)
2 - FRONT - MIDDLE
3 - FRONT - RIGHT SIDE
4 - SECOND - LEFT SIDE (MOTORCYCLE PASSENGER)
5 - SECOND - MIDDLE
6 - SECOND - RIGHT SIDE
7 - THIRD - LEFT SIDE (MOTORCYCLE SIDE CAR)
8 - THIRD - MIDDLE
9 - THIRD - RIGHT SIDE
10 - SLEEPER SECTION OF TRUCK CAB
11 - PASSENGER IN OTHER ENCLOSED CARGO AREA (NON-TRAILING UNIT, BUS, PICK-UP WITH CAP)
12 - PASSENGER IN UNENCLOSED CARGO AREA
13 - TRAILING UNIT
14 - RIDING ON VEHICLE EXTERIOR (NON-TRAILING UNIT)
15 - NON-MOTORIST
99 - OTHER / UNKNOWN

AIR BAG

1 - NOT DEPLOYED
2 - DEPLOYED FRONT
3 - DEPLOYED SIDE
4 - DEPLOYED BOTH FRONT / SIDE
5 - NOT APPLICABLE
9 - DEPLOYMENT UNKNOWN

EJECTION

1 - NOT EJECTED
2 - PARTIALLY EJECTED
3 - TOTALLY EJECTED
4 - NOT APPLICABLE

TRAPPED

1 - NOT TRAPPED
2 - EXTRICATED BY MECHANICAL MEANS
3 - FREED BY NON-MECHANICAL MEANS

OL CLASS

1 - CLASS A
2 - CLASS B
3 - CLASS C
4 - REGULAR CLASS (OHIO = D)
5 - M/C MOPED ONLY
6 - NO VALID OL

OL ENDORSEMENT

H - HAZMAT
M - MOTORCYCLE
P - PASSENGER
N - TANKER
Q - MOTOR SCOOTER
R - THREE-WHEEL MOTORCYCLE
S - SCHOOL BUS
T - DOUBLE & TRIPLE TRAILERS
X - TANKER / HAZMAT

GENDER

F - FEMALE
M - MALE
U - OTHER/UNKNOWN

OL RESTRICTION(S)

1 - ALCOHOL INTERLOCK DEVICE
2 - CDL INTRASTATE ONLY
3 - CORRECTIVE LENSES
4 - FARM WAIVER
5 - EXCEPT CLASS A BUS
6 - EXCEPT CLASS A & CLASS B BUS
7 - EXCEPT TRACTOR-TRAILER
8 - INTERMEDIATE LICENSE RESTRICTIONS
9 - LEARNER'S PERMIT RESTRICTIONS
10 - LIMITED TO DAYLIGHT ONLY
11 - LIMITED TO EMPLOYMENT
12 - LIMITED - OTHER
13 - MECHANICAL DEVICES (SPECIAL BRAKES, HAND CONTROLS, OR OTHER ADAPTIVE DEVICES)
14 - MILITARY VEHICLES ONLY
15 - MOTOR VEHICLES WITHOUT AIR BRAKES
16 - OUTSIDE MIRROR
17 - PROSTHETIC AID
18 - OTHER

DRIVER DISTRACTION

1 - NOT DISTRACTED
2 - MANUALLY OPERATING AN ELECTRONIC COMMUNICATION DEVICE (TEXTING, TYPING, DIALING)
3 - TALKING ON HANDS-FREE COMMUNICATION DEVICE
4 - TALKING ON HAND-HELD COMMUNICATION DEVICE
5 - OTHER ACTIVITY WITH AN ELECTRONIC DEVICE
6 - PASSENGER
7 - OTHER DISTRACTION INSIDE THE VEHICLE
8 - OTHER DISTRACTION OUTSIDE THE VEHICLE
9 - OTHER / UNKNOWN

CONDITION

1 - APPARENTLY NORMAL
2 - PHYSICAL IMPAIRMENT
3 - EMOTIONAL (E.G., DEPRESSED, ANGRY, DISTURBED)
4 - ILLNESS
5 - FELL ASLEEP, FAINTED, FATIGUED, ETC.
6 - UNDER THE INFLUENCE OF MEDICATIONS / DRUGS / ALCOHOL
9 - OTHER / UNKNOWN

TEST STATUS

1 - NONE GIVEN
2 - TEST REFUSED
3 - TEST GIVEN, CONTAMINATED SAMPLE / UNUSABLE
4 - TEST GIVEN, RESULTS KNOWN
5 - TEST GIVEN, RESULTS UNKNOWN

ALCOHOL TEST TYPE

1 - NONE
2 - BLOOD
3 - URINE
4 - BREATH
5 - OTHER

DRUG TEST TYPE

1 - NONE
2 - BLOOD
3 - URINE
4 - OTHER

DRUG TEST RESULT(S)

1 - AMPHETAMINES
2 - BARBITURATES
3 - BENZODIAZEPINES
4 - CANNABINOIDS
5 - COCAINE
6 - OPIATES / OPIOIDS
7 - OTHER
8 - NEGATIVE RESULTS

I have tried many python libraries like *PyPDF2*, *PDFMiner*, *pikepdf*, Camelot, and *tabulat*. However, none of them worked except *PyMuPDF*.

Before going into the code it's important to understand the meaning of 2 important terms which would help in understanding the code.

> Word: Sequence of characters without space. Ex – ash, 23, 2, 3.
>
> Annots: An annotation associates an object such as a note, image, or bounding box with a location on a page of a PDF document, or provides a way to interact with the user using the mouse and keyboard. The objects are called annots.

Please note that in our case the bounding box, annots, and rectangles are the same thing. Therefore, these terms would be used interchangeably.

First, we will extract text from one of the bounding boxes. Then we will use the same procedure to extract data from all the bounding boxes of pdf.

Code:

```
import fitz import pandas as pd doc = fitz.open('Mansfield--70-21009048 - ConvertToExcel.pdf') page1 = doc[0]
words = page1.get_text("words")
```

Firstly, we import the *fitz* module of the *PyMuPDF* library and pandas library. Then the object of the PDF file is created and stored in doc and 1st page of pdf is stored on page1. *page.get_text()* extracts all the words of page 1. Each word consists of a tuple with 8 elements.

In words variable, the First 4 elements represent the coordinates of the word, 5th element is the word itself, 6th,7th, 8th elements are block, line, word numbers respectively.

OUTPUT

Extract the coordinates of the first object :

```
first_annots=[] rec=page1.first_annot.rect rec #Information of words in first object is stored in mywords
mywords = [w for w in words if fitz.Rect(w[:4]) in rec] ann= make_text(mywords) first_annots.append(ann)
```

This function selects the words contained in the box, sort the words and return in form of a string :

```
def make_text(words): line_dict = {} words.sort(key=lambda w: w[0]) for w in words: y1 = round(w[3], 1) word
= w[4] line = line_dict.get(y1, []) line.append(word) line_dict[y1] = line lines = list(line_dict.items())
lines.sort() return "n".join([" ".join(line[1]) for line in lines])
```

OUTPUT

page.first_annot() gives the first annot i.e. bounding box of the page.

.rect gives coordinates of a rectangle.

Now, we got the coordinates of the rectangle and all the words on the page. We then filter the words which are present in our bounding box and store them in *mywords* variable.

We have got all the words in the rectangle with their coordinates. However, these words are in random order. Since we need the text sequentially and that only makes sense, we used a function make_text() which first sorts the words from left to right and then from top to bottom. It returns the text in string format.

Hurrah! We have extracted data from one annot. Our next task is to extract data from all annots of the PDF which would be done in the same approach.

Extracting each page of the document and all the annots/rectanges :

```
for pageno in range(0,len(doc)-1): page = doc[pageno] words = page.get_text("words") for annot in
page.annots(): if annot!=None: rec=annot.rect mywords = [w for w in words if fitz.Rect(w[:4]) in rec] ann=
make_text(mywords) all_annots.append(ann)
```

all_annots, a list is initialized to store the text of all annots in the pdf.

The function of the outer loop in the above code is to go through each page of PDF, while that of the inner loop is to go through all annots of the page and performing the task of adding texts to all_annots list as discussed earlier.

Printing all_annots provides us the text of all annots of the pdf which you can see below.

OUTPUT

Finally, we have extracted the texts from all the annots/ bounding boxes.

Its time to clean the data and bring it in an understandable form.

Data Cleaning and Data Processing

Splitting to form column name and its values :

```
cont=[] for i in range(0,len(all_annots)): cont.append(all_annots[i].split('n',1))
```

```
-
```

Removing unnecessary symbols *,#,:

```
liss=[] for i in range(0,len(cont)): lis=[] for j in cont[i]: j=j.replace('*','') j=j.replace('#','')
j=j.replace(':','') j=j.strip() #print(j) lis.append(j) liss.append(lis)
```

Spliting into keys and values and removing spaces in the values which only contain digits :

```
keys=[] values=[] for i in liss: keys.append(i[0]) values.append(i[1]) for i in range(0, len(values)): for j
in range(0,len(values[i])): if values[i][j]>='A' and values[i][j]<='Z': break if j==len(values[i])-1:
values[i]=values[i].replace(' ','')
```

We split each string based on a new line (n) character to separate the column name from its values. By further cleaning unnecessary symbols like (*, #, □ are removed. Spaces between digits are removed.

With the key-value pairs, we create a dictionary which is shown below:

Converting to dictionary :

```
report=dict(zip(keys,values))
```

report[‘VEHICLE IDENTIFICATION’]=report[‘VEHICLE IDENTIFICATION’].replace(‘ ‘,”)

```
dic=[report['LOCALITY'],report['MANNER OF CRASH COLLISION/IMPACT'],report['CRASH SEVERITY']] l=0 val_after=[]
for local in dic: li=[] lii=[] k='' extract='' l=0 for i in range(0,len(local)-1): if local[i+1]>='0' and
local[i+1]<='9': li.append(local[l:i+1]) l=i+1 li.append(local[l:]) print(li) for i in li: if i[0] in lii:
k=i[0] break lii.append(i[0]) for i in li: if i[0]==k:
```

extract=i

```
val_after.append(extract) break report['LOCALITY']=val_after[0] report['MANNER OF CRASH
COLLISION/IMPACT']=val_after[1] report['CRASH SEVERITY']=val_after[2]
```

OUTPUT

Lastly, dictionary is converted to dataframe with the help of pandas.

Converting to DataFrame and exporting to CSV:

```
data=pd.DataFrame.from_dict(report)



data.to_csv('final.csv',index=False)
```

OUTPUT

Now, we can perform analysis on our structured data or export it to excel.

I hope that you have enjoyed reading this blog and it has given you an intuition of dealing with unstructured data.

References:

Source of the featured image: Real Python https://realpython.com/python-data-engineer/

PyMuPDF documentation : https://pymupdf.readthedocs.io/en/latest/

About the Author:

Hi! I am Ashish Choudhary. I am pursuing B.Tech from the JC Bose University of Science & Technology. Data Science is my passion and feels proud to write interesting blogs related to it. Feel free to contact me on Linkedin linkedin.com/in/ashish-choudhary-7b6029166.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

Article Url - https://www.analyticsvidhya.com/blog/2021/06/data-extraction-from-unstructured-pdfs/

ashish@choudhary