PharmaSUG 2024 - Paper SD-370

Enhancing FDA Debarment List Compliance through Automated Data Analysis Using Python and SAS

Yongjiang (Jerry) Xu, Genmab Inc.
Karen Xu, Northeastern University
Suzanne Viselli, Genmab Inc.

ABSTRACT

This paper presents an innovative approach to ensuring compliance with the FDA's debarment list by employing Python and SAS for automated data analysis. The primary objective was to develop and streamline the process of checking individuals and entities against the FDA's debarment list, thereby reducing the risk of regulatory violations. We utilized Python for data scraping and extracting debarment data from the FDA's website, followed by SAS for data analysis and provided a comprehensive compliance solution. The results demonstrated a significant reduction in manual labor and time required for compliance checks, with the added benefit of minimizing human error. The scalable and efficient tool automatically flags if any individuals or entities are on the debarment list from international trials, facilitating timely decision-making.

Key words:

FDA Debarment List, Regulatory Compliance, Python, SAS, Fuzzy Matching Programming

INTRODUCTION

In the pharmaceutical and healthcare industries, compliance with regulatory standards is not just a legal imperative but a cornerstone of ethical operations. One critical aspect of this compliance involves adhering to the United States Food and Drug Administration's (FDA) debarment list. The FDA debarment list enumerates individuals and entities prohibited from participating in any capacity in the drug product industry, primarily due to violations of the Food, Drug, and Cosmetic Act (FDCA). This list plays a pivotal role in safeguarding public health and maintaining the integrity of the pharmaceutical supply chain. However, the process of ensuring that organizations do not engage with debarred parties can be intricate and labor-intensive, particularly for clinical activities with extensive personnel and partnerships.

An automation compliance plan is developed by the collaboration of regulatory and programming teams. A knowledge sharing collaboration of SAS programming team with new generation knowing computer language detailed the implementation plan. Recent advancements in data analytics and automation offer a promising avenue for enhancing compliance procedures. In this context, we explore the integration of Python and SAS to develop an automated system for FDA debarment list compliance. Python, known for its versatility and robust data handling capabilities, is utilized for extracting and preprocessing debarment data from the FDA's website. On the other hand, SAS provides sophisticated data analysis and reporting functionalities, crucial for managing and interpreting complex datasets within organizational frameworks. The paper is to demonstrate how the synergy between Python and SAS can be harnessed to create an efficient, accurate, and user-friendly tool for debarment list compliance. This tool aims to significantly reduce the manual effort and potential human error inherent in traditional compliance checks.

SCRAPE THE DEBARMENTLIST FROM FDA WEBSITE

DEBARMENT LIST

The version of debarment list used in the paper is dated 12/01/2023. It is in FDA's website (https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/compliance-

actions-and-activities/fda-debarment-list-drug-product-applications). As requested by the regulatory team, the expired debarment list has also been scraped and checked against. The expired debarment list is on the website below. https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/fda-debarment-list-drug-product-applications/fda-expired-debarment-list-drug-product-applications. The paper illustrates the process using the current debarment list.

Since SAS doesn't natively support scraping data from web pages, an undergraduate student used a Python script to data-scrape from FDA's website. The following example was written using Python 3 and the Pandas library.

```
#import necessary libraries
import pandas as pd
import numpy as py

# use pandas.read_html to read all tables from the given URL
all_tables = pd.read_html('https://www.fda.gov/inspections-compliance-
enforcement-and-criminal-investigations/compliance-actions-and-
activities/fda-debarment-list-drug-product-applications')

# there are two tables on the website
# we only want the second one (Persons) as the first one is empty
# so, we should define a new variable that only has the second table
persons_table = all_tables[1]

#check that the type of the persons_table is indeed a DataFrame
print(type(persons_table))

#take a look at the DataFrame
print(persons_table)

# In expired debarment list, the first line appears to be a duplicate
# of the column names. It's been removed by code below it
persons_table = persons_table.drop(0, axis = 0);

# export it to a csv file
persons_table.to_csv("Persons.csv")
```

Below is a snip of the resulting file.

	A	B	C	D	E	F	G
1		Last Name	First & Middle Names	Effective Date	End/Term of Debarment	FR Date.txt (MM/DD/YY)	Volume Page.pdf
2	1	Akhigbe	Ehigiator O.	12/17/2010	25 Year%	12/17/2010	75 FR 79005
3	2	Albanese	Anthony W.	11/23/2009	Permanent^	11/23/2009	74 FR 61151

DATA PREPARATION OF THE DEBARMENT LIST

After the debarment list in csv file is created, further data preparation is carried out using SAS.

Step 1. In debarment list, the first name and middle name are in one single column. To facilitate the check, we split them into two columns, one for the first name and the other for the middle name. In later process, last name and first name have been compared, respectively.

```
****Read in data****;
proc import datafile = "&persons"
```

```
                out = per
                dbms = csv replace;
                GETNAMES=YES;
                DATAROW=2;

        run;
```

A snip of dataset per:

VAR1	Last_Name	First___Middle_Names	Effective_Date	End_Term_of_Debarment	FR_Date_txt___MM_DD_YY_	Volume_Page_pdf
1	Akhigbe	Ehigiator O.	12/17/2010	25 Year%	12/17/2010	75 FR 79005
2	Albanese	Anthony W.	11/23/2009	Permanent^	11/23/2009	74 FR 61151

```
        ****Clean data****;

        data debar1;
         set  debar;
         length fda_frst fda_mid  fda_last $200;
                fda_frst = scan (first___middle_names, 1, ' ');
                fda_mid= scan ( first___middle_names, 2, ' ');
                fda_last= strip (last_name);
         run;
```

Step 2. A few of last names included the a.k.a (also known as) part, which are another version of full names. They were also moved to additional rows, in which the last, first, and middle names were extracted.

```
        data debar2;
        set debar1;
        length fda_last1 last2  $200;
                fda_last=tranwrd(upcase(fda_last), 'A.K.A.', '|'); ;

                if index(fda_last, '|') then do;
                        fda_last1=compress(strip(scan(fda_last, 1, '|')), ',');
                        last2=strip(scan(fda_last, -1, '|'));
                end;
                else do;
                        fda_last1=fda_last;
                        last2='';
                end;

                output;

                if last2 ne '' then do;
                        if index(last2, ',') then do;
                                fda_last1=strip(scan(last2, 1, ','));
                                fda_frst=strip(scan(last2, -1, ','));
                        end;
                        else do;
                                fda_last1=strip(scan(last2, -1, ' '));
                                fda_frst=strip(scan(last2, 1, ' '));
                        end;
                output;
                end;
         run;
```

FUZZY MATCHING USING SOUNDEX AND COMPGED

The clinical investigators' data (first name, last name, middle names, site name and address, contact information, etc.) in Excel was provided by the clinical site management team. This data were read into SAS data set MIL by proc import.

To enhance flexibility and accuracy in matching names that are either exact same, or not exact but are similar in sound or composition, the approach applied is fuzzy match techniques which was discussed in a paper by Sloan and Lafler (Sloan and Lafler, PharmaSUG2022 – Paper AP-030). The SOUNDEX and COMPGED were selected in our work.

SOUNDEX function converts a string to a code based on how it sounds when spoken. Strings that sound similar will have the same SOUNDEX code. It's useful for finding names that might be spelled differently but sound alike. Below is the sample code to find matches of SOUNDEX codes of first names and last names from both datasets debar2 and mil.

```
Proc sql ;
create table mtch_sd as
Select a.*,
      b.*,
      soundex(a.fda_frst) as sound1a,
      soundex(b.first_name) as sound1b,
      soundex(a.fda_last1) as sound2a,
      soundex(b.last_name) as sound2b

From debar2 as a,
      mil as b
      Where (a.fda_last1 ne '') and ( ( calculated sound1a=calculated
      sound1b   )  and (calculated sound2a = calculated sound2b))
Order by a.fda_last1;
quit ;
```

COMPGED: The COMPGED function computes the generalized edit distance between two strings. This distance is a measure of how many operations (like insertions, deletions, or substitutions) are needed to transform one string into another. It's useful for matching strings that are the same or similar but not exactly the same. Below is the sample code of COMPGED function with a modifier value of "INL" to compare first names and last names from both datasets debar2 and mil. 'INL' is to ignore the case, remove leading blanks, and ignore quotes around corresponding variables. The "cutoff-value" for the COMPGED_Score is set at 100 as most COMPGED scores of 100 or less are valid matches for the comparison that they are performing (Sloan and Lafler, PharmaSUG2022 – Paper AP-030) (Cadieux and Bretheim, paper 1674-2014, 2014).

```
Proc sql ;
create table mtch_cd as
Select a.*,
            b.*,
      COMPGED(a.fda_frst, b.first_name, 'INL') AS score1,
      COMPGED(a.fda_last1, b.last_name, 'INL') AS score2

From debar as a,
      mil as b
      Where (a.fda_last1 ne '') and calculated score1 le 100 and
      calculated score2 le 100
Order by a.fda_last1;
quit ;
```

To cast a wide net of possible matching, the paper did not include the middle name checks as the importance of a middle name is subjective and varies widely. However, from a legal standpoint, it can be an essential part of identifying information, especially in contexts requiring unambiguous identification.

OUTPUT THE RESULT FOR REGULATORY REVIEW

To be conservative and cautious, the match records, if any, from SOUNDEX and COMPGED will be output to Excel and further assessed by Regulatory. In addition, a note of 'No match was found between the PIs/sub-PIs and debarment list from FDA' will display if there is no match record.

Instead of spending days to check thousands of clinical principal and sub-principal investigators' names against FDA's debarment list, regulatory team now only checked a couple of lines in several minutes in the study case.

```
**** The following records need regulatory review****;
data chk;
set mtch_sd mtch_cd;

run;

proc sql noprint;
    select count(*) into :rcrdn from chk;
quit;

%if &rcrdn = 0 %then %do;
    data chk;
      length note $200;
      note='No match was found between the PIs/sub-PIs and debarment
lists from FDA' ;
      run;
%end;

****Output****;
ods listing close;
ods excel file="&output_path.\similar_debar.xlsx" style = normal
  options ( FLOW='Tables' embedded_titles="yes" frozen_headers="3" );

ods excel options (sheet_name = "Listing"
   tab_color='grey'  orientation = "landscape");
proc print data=chk  noobs label ;

run;

ods excel close;
ods listing ;
```

CONCLUSION

A practical solution was introduced for automating compliance checks against the FDA's debarment list using Python for data scraping and SAS for analysis. The approach, by combination of Python and SAS, significantly reduces the manual effort and time traditionally required for such checks, minimizes human error, and enhances the efficiency and scalability of the compliance process. The developed tool supports more timely and accurate compliance decisions, demonstrating a valuable contribution to regulatory adherence practices.

REFERENCES

Sloan, Stephen and Kirk Paul Lafler (2022). “A Quick Look at Fuzzy Matching Programming

Techniques Using SAS® Software.” *Proceedings of the 2022 PharmaSUG Conference*.

Cadieux, Richard and Daniel R. Brethiem (2014). “Matching Rules: Too Loose, Too Tight, or Just Right?”, *Proceedings of the 2014 SAS Global Forum (SGF) Conference*.

U.S. Foos & Drug Administration. “FDA Debarment List (Drug Product Applications)”. 12/01/2023. Available at https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/compliance-actions-and-activities/fda-debarment-list-drug-product-applications

U.S. Foos & Drug Administration. “FDA Expired Debarment List (Drug Product Applications)”. 12/01/2023. Available at https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/fda-debarment-list-drug-product-applications/fda-expired-debarment-list-drug-product-applications

ACKNOWLEDGMENTS

We would like to acknowledge our Regulatory colleagues Ivy Nyarko and Layne Chaya for the collaboration. We would also like to thank Genmab’s programming leadership team (Chris Velas, Suzanne Viselli, Jennifer Gu, Justin Boland and Jan Skowronski) for the support and endorsement.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Yongjiang (Jerry) Xu
Genmab Inc.
E-mail: jexu@genmab.com

Karen Xu
Northeastern University, Class of 2024
BS in Computer Science and Business Administration
E-mail: karenxu27@gmail.com

Suzanne Viselli
Genmab Inc.
E-mail: suvi@genmab.com