Capstone Project: CRM DATA CLEANSING

Sponsoring Organization: Berkadia

Sponsoring Individual with Title: Gary Mumford, Vice President of Data Governance

Sponsoring Individual Phone & Email: Gary.Mumford@berkadia.com

Faculty Advisor with Email: Professor Rohit Aggarwal, rohit.aggarwal@eccles.utah.edu

Team Members: Abhishek Anney, Bhuvananjali Challagalla, Sana Kanwar, Sukrit Sen

Table of Contents

Executive Summary

Berkadia is a joint venture of Berkshire Hathaway and Leucadia National Corporation
(now Jefferies Financial Group) – both of which are renowned for their capital strength and sophisticated investment strategies that mostly deal with multi-property and commercial property clients. It is amongst the largest, highest-rated and most respected primary, master, and special services in the industry.

The company deals with huge customer data which needs proper management, So as to help the sales and management team to be more effective and increase their revenue. The company does all the verification and validation work manually. Which as one can imagine, this is very much slow and time-consuming to go through all the customer data manually to find matching records and to validate it, and this also increases the scope for error as there are high chances that one might overlook some records. Therefore, Berkadia, as our sponsor, has requested the development of an automated model, using Python and Machine Learning libraries of Python, that would clean the data by removing duplicate entries and clean the data whenever data is passed into it.

The whole process of building the model is done using the Python programming language. The models that are built will pre-process the data and normalize the data to a stage where we can apply ML algorithms and clean the data for further analysis. The algorithm would divide the data into clean data and redundant data when passed into it. Several models were built such as the support vector machine model, Decision Tree model for training and testing the data and finally, we decided to go ahead with the random forest model. The Machine Learning Algorithms created to achieve these results are calculation-intensive and require high processing power for the machine.

Project Objectives and Requirements

Overview:

●People move often to different roles within the organization or join a different organization and an approximate percentage change of 2% per month and 25% per year is observed by Berkadia. The project is inclined towards working on refining data of CRM database which has contact and account information. This data inaccuracy does not benefit Berkadia from a sales point of view.

●The Primary problem of the dataset is duplicate entries.

●We will not be directly connected to the CRM database instead we would be given data in CSV or Excel format.

Business Problem:

- Slow and time-consuming
- Increased Error of chances
- Inefficient and inaccurate
- Data Analysis is not leveraged completely on the data-set.

Strategic Limitation:

- Rapid decay of CRM data captured on Salesforce.
- Frequent change in Customer Information
- (2% per month & 25% per year).
- Effective utilization of Sales and Marketing Teams.

Milestones:

Project Working Hours: 20 Hours/Week

Milestones	Timeline (Expected)
Initial Sampling	End of January 2019
Research to List the algorithms	End of February 2019
Validate the selection Process	Mid-March 2019
Finalize an Algorithm - Validation, and Testing	Mid-April 2019
Implementation	May-June 2019
Project Presentation	July 2019

Risks and Assumptions:

Since this is a project containing sensitive data information, there are several risks that must be addressed. There is a cascade effect on the risks where each factor affects the bottom line. This is a big data project and the size of the data could be a risk factor impacting the lack of an infrastructure tool to deal with larger datasets.

Pre-Completion:

- Meeting set-up & Meeting dynamics will be discussed as per availability and convenience,
- Statistical Sampling should be done with complete ownership.
- Training will not be imparted and learning by students will be driving the project. Students will be taking advice from the Faculty Mentor.

Post-Completion:

Higher configuration of the machine is required to run the algorithm completely. Multiple Batch Processing - Higher Run-time.Minimum configuration for testing- I5 with 12GB RAM and 2.7 GHz Processor. [GPU Will be preferable].

Project Workflow and EDA:

- Thorough understanding of the data dictionary and the different methods that can be used to clean the data and perform necessary Initial Data Preprocessing (Normalization and Cleaning.

- Merged the datasets and worked on Identifying variables that make the best of a model.

- Choose the fields that can be used in the data de-duplication phase and hence create a randomly sampled test data-set for validation and experimentation of the process designed.

- After the identification of the fields and the consultation with the Berkadia Team and Faculty Advisor, the next plan was to choose and develop an appropriate methodology for cleaning the data.

- Generating Machine Learning Algorithms that will suffice the requirement of the process.

- Training the ML Algorithm to find duplicate data and then executing the appropriate model on the data-set based on the accuracy achieved in the test data-set.

- Segregating the cleaned data file and separating the redundant along with creating a model metric dashboard which measures the performance based on the fuzzy scores.

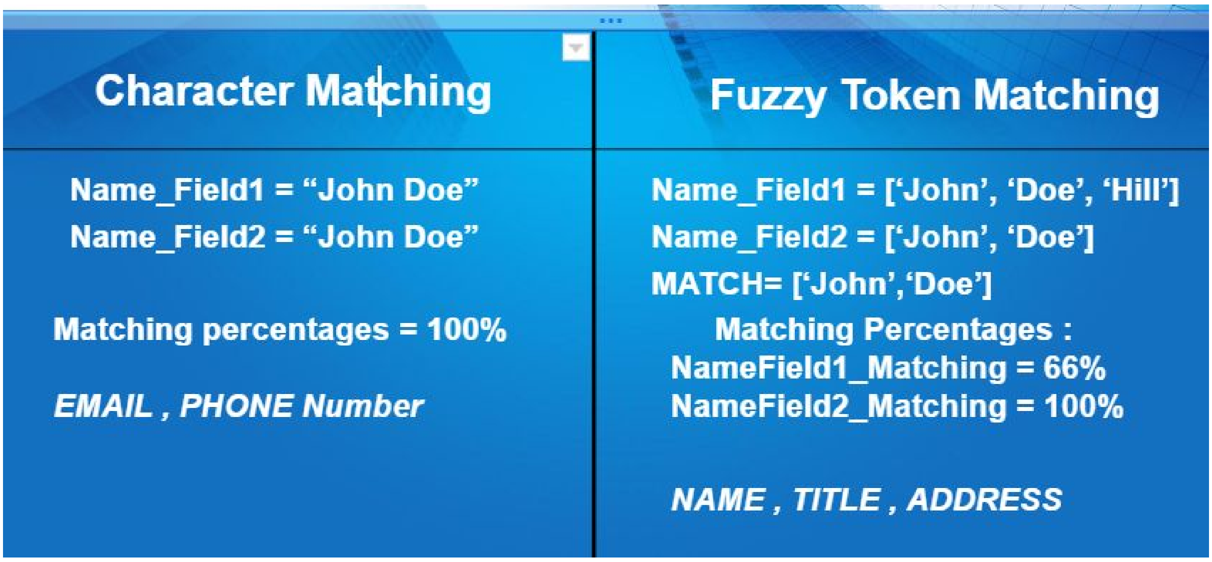
- Saving the models for future prediction and implementing the models on the testing data-set and finding the accuracy of the project and the models.

- Providing the deliverables to the client and informing the risks and assumptions. A technical document for walkthrough with the project files and code will also be provided to the client.

ANALYSIS AND MODEL METRICS:

We have used the concept of fuzzy matching which is used in finding correspondences between segments of a text and entries in a database using the ratio of similarity score. The variables chosen for this phase are as follows:

- ❖ *Name Matching Percentages.*
- ❖ *Email Matching Percentages.*
- ❖ *Phone Matching Percentages.*
- ❖ *Title Matching Percentages.*
- ❖ *Company Name Matching Percentages.*
- ❖ *Address Matching Percentages.*



Character Matching

Name_Field1 = "John Doe"
Name_Field2 = "John Doe"

Matching percentages = 100%

EMAIL , PHONE Number

Fuzzy Token Matching

Name_Field1 = ['John', 'Doe', 'Hill']
Name_Field2 = ['John', 'Doe']
MATCH= ['John','Doe']

Matching Percentages :
NameField1_Matching = 66%
NameField2_Matching = 100%

NAME , TITLE , ADDRESS

MODEL DEVELOPMENT AND PROCESS FLOW:

•Creating Matrices using above found percentages and training ML Algorithm model (SVM, Decision Tree).

•After we received the percentages, we selected machine learning models that can handle ratio of similarity or the percentage as an input and provides the o/p in a format that can be implemented on the complete data-set.

•Models that were selected for this process is Support Vector Machine and Random Forest.

•The target classification label was prepared, which was further used for the training the Support Vector Machine Algorithm.

•Random Forest was chosen as it is one of the most highly effective ML algorithms and it produces a highly accurate classifier on large data-sets, which is our case since our data-set has around 200K rows.

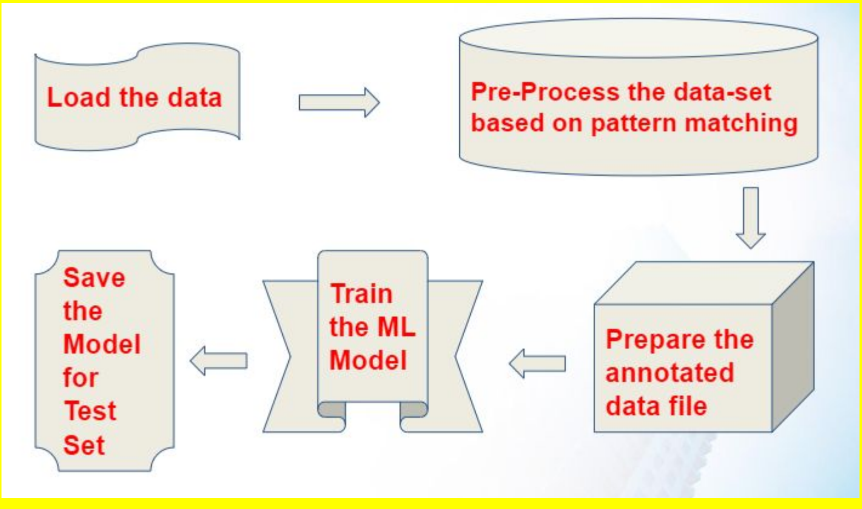
Percentages ↠↠↠↠↠↞◇↠↠↠↠ Numpy Array of features.

Data Annotation: Labeled data for Machine Learning Algorithm training.

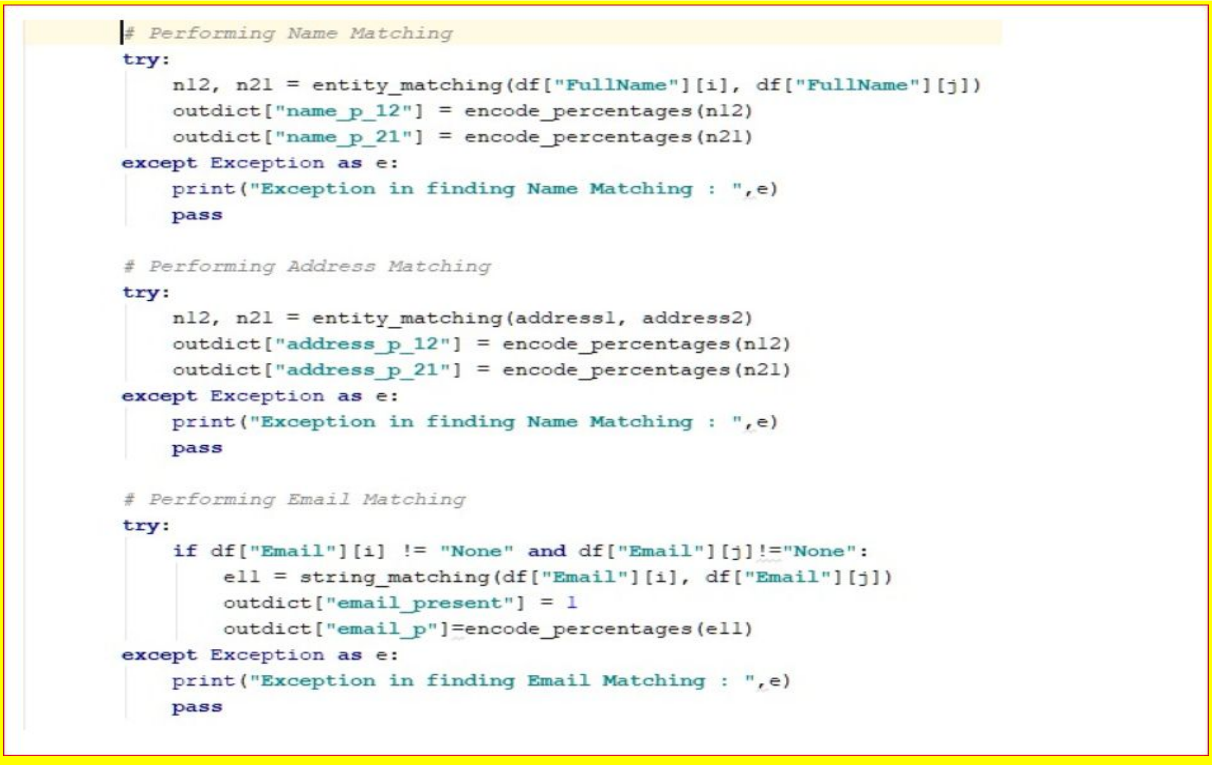
Train Test Split.

Predicting and Validating Test Values.

Saving models for Future Predictions.


Load the data
Pre-Process the data-set based on pattern matching
Prepare the annotated data file
Train the ML Model
Save the Model for Test Set

Data Pre-Processing:

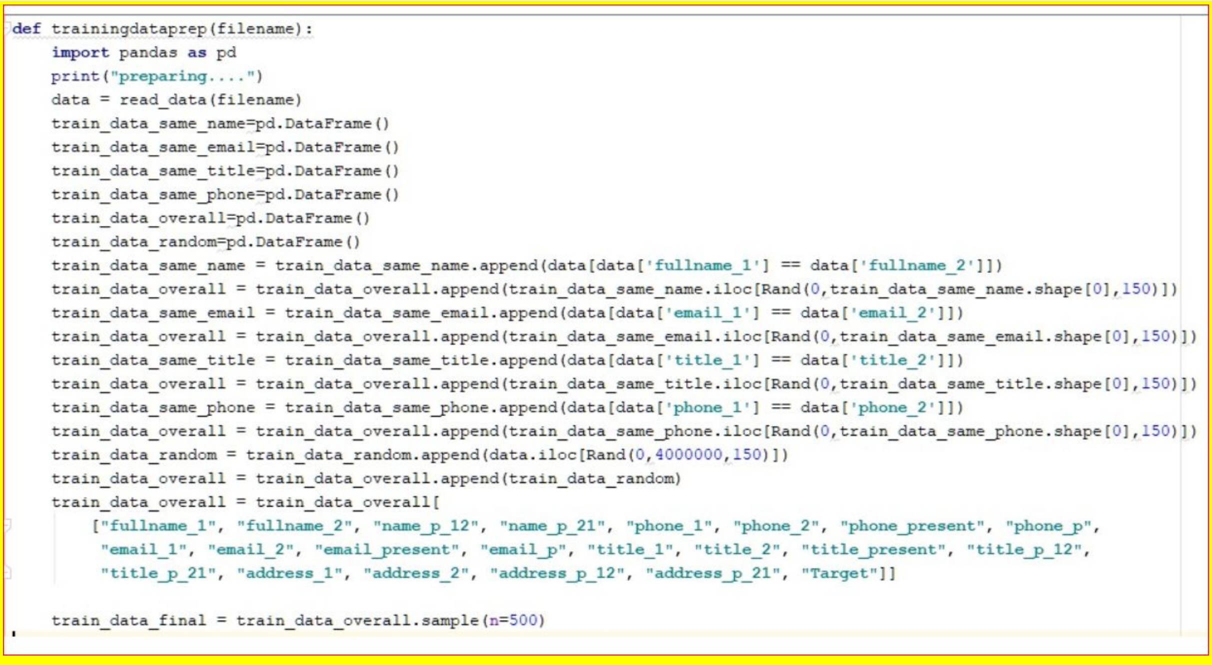


```
# Performing Name Matching
try:
    n12, n21 = entity_matching(df["FullName"][i], df["FullName"][j])
    outdict["name_p_12"] = encode_percentages(n12)
    outdict["name_p_21"] = encode_percentages(n21)
except Exception as e:
    print("Exception in finding Name Matching : ",e)
    pass

# Performing Address Matching
try:
    n12, n21 = entity_matching(address1, address2)
    outdict["address_p_12"] = encode_percentages(n12)
    outdict["address_p_21"] = encode_percentages(n21)
except Exception as e:
    print("Exception in finding Name Matching : ",e)
    pass

# Performing Email Matching
try:
    if df["Email"][i] != "None" and df["Email"][j]!="None":
        e11 = string_matching(df["Email"][i], df["Email"][j])
        outdict["email_present"] = 1
        outdict["email_p"]=encode_percentages(e11)
except Exception as e:
    print("Exception in finding Email Matching : ",e)
    pass
```

Model Training:



```
def trainingdataprep(filename):
    import pandas as pd
    print("preparing....")
    data = read_data(filename)
    train_data_same_name=pd.DataFrame()
    train_data_same_email=pd.DataFrame()
    train_data_same_title=pd.DataFrame()
    train_data_same_phone=pd.DataFrame()
    train_data_overall=pd.DataFrame()
    train_data_random=pd.DataFrame()
    train_data_same_name = train_data_same_name.append(data[data['fullname_1'] == data['fullname_2']])
    train_data_overall = train_data_overall.append(train_data_same_name.iloc[Rand(0,train_data_same_name.shape[0],150)])
    train_data_same_email = train_data_same_email.append(data[data['email_1'] == data['email_2']])
    train_data_overall = train_data_overall.append(train_data_same_email.iloc[Rand(0,train_data_same_email.shape[0],150)])
    train_data_same_title = train_data_same_title.append(data[data['title_1'] == data['title_2']])
    train_data_overall = train_data_overall.append(train_data_same_title.iloc[Rand(0,train_data_same_title.shape[0],150)])
    train_data_same_phone = train_data_same_phone.append(data[data['phone_1'] == data['phone_2']])
    train_data_overall = train_data_overall.append(train_data_same_phone.iloc[Rand(0,train_data_same_phone.shape[0],150)])
    train_data_random = train_data_random.append(data.iloc[Rand(0,4000000,150)])
    train_data_overall = train_data_overall.append(train_data_random)
    train_data_overall = train_data_overall[
        ["fullname_1", "fullname_2", "name_p_12", "name_p_21", "phone_1", "phone_2", "phone_present", "phone_p",
         "email_1", "email_2", "email_present", "email_p", "title_1", "title_2", "title_present", "title_p_12",
         "title_p_21", "address_1", "address_2", "address_p_12", "address_p_21", "Target"]]

    train_data_final = train_data_overall.sample(n=500)
```

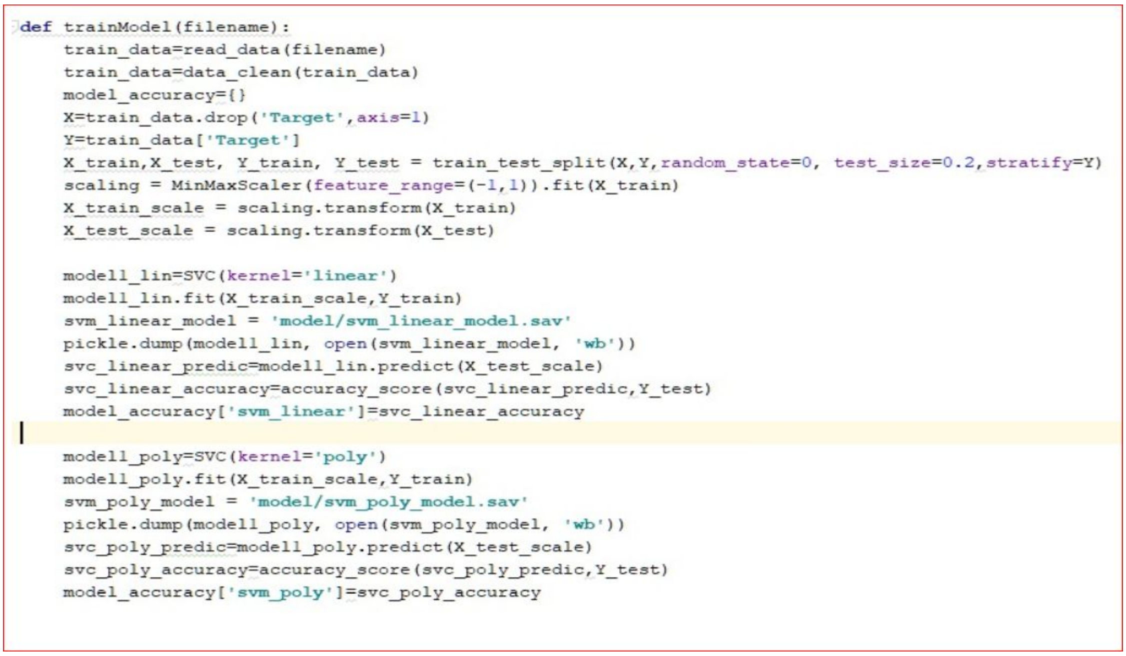
Model Selection and Comparison:

Support Vector Machine:

The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space(N — the number of features) that distinctly classifies the data points.

Random Forest:

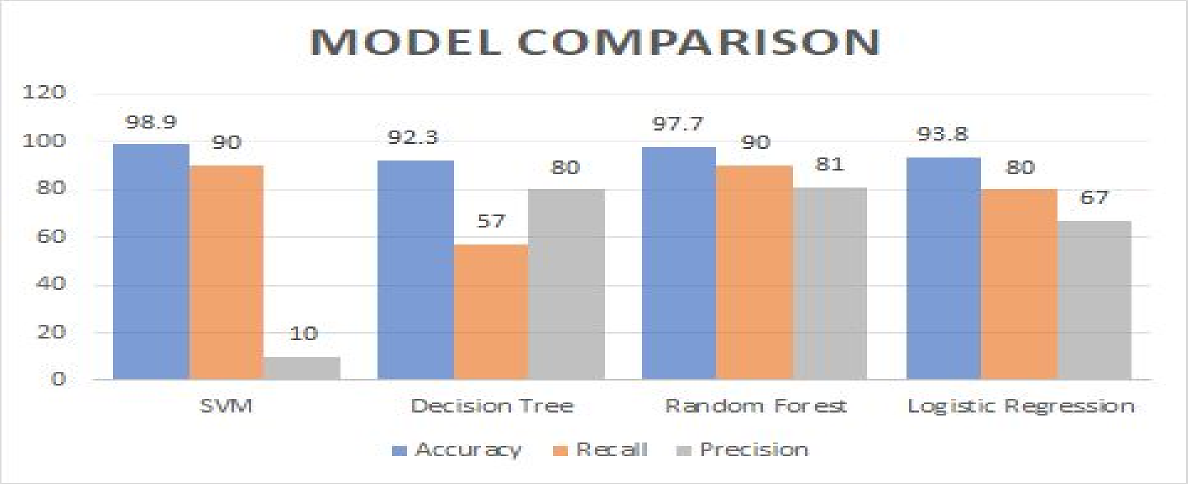
A random forest consists of multiple random decision trees. Two types of randomnesses are built into the trees. First, each tree is built on a random sample from the original data. Second, at each tree node, a subset of features are randomly selected to generate the best split.



```
def trainModel(filename):
    train_data=read_data(filename)
    train_data=data_clean(train_data)
    model_accuracy={}
    X=train_data.drop('Target',axis=1)
    Y=train_data['Target']
    X_train,X_test, Y_train, Y_test = train_test_split(X,Y,random_state=0, test_size=0.2,stratify=Y)
    scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
    X_train_scale = scaling.transform(X_train)
    X_test_scale = scaling.transform(X_test)

    modell_lin=SVC(kernel='linear')
    modell_lin.fit(X_train_scale,Y_train)
    svm_linear_model = 'model/svm_linear_model.sav'
    pickle.dump(modell_lin, open(svm_linear_model, 'wb'))
    svc_linear_predic=modell_lin.predict(X_test_scale)
    svc_linear_accuracy=accuracy_score(svc_linear_predic,Y_test)
    model_accuracy['svm_linear']=svc_linear_accuracy

    modell_poly=SVC(kernel='poly')
    modell_poly.fit(X_train_scale,Y_train)
    svm_poly_model = 'model/svm_poly_model.sav'
    pickle.dump(modell_poly, open(svm_poly_model, 'wb'))
    svc_poly_predic=modell_poly.predict(X_test_scale)
    svc_poly_accuracy=accuracy_score(svc_poly_predic,Y_test)
    model_accuracy['svm_poly']=svc_poly_accuracy
```



TESTING AND VALIDATION:

•Validating Test Data and Accuracy.

•Finding Duplicate Entries along with Confidence and Threshold Score.

•These confidence and threshold scores are generated through the ML algorithms and hence we can create 2 separate files which is one for cleaned data and another for a compilation of redundant data.

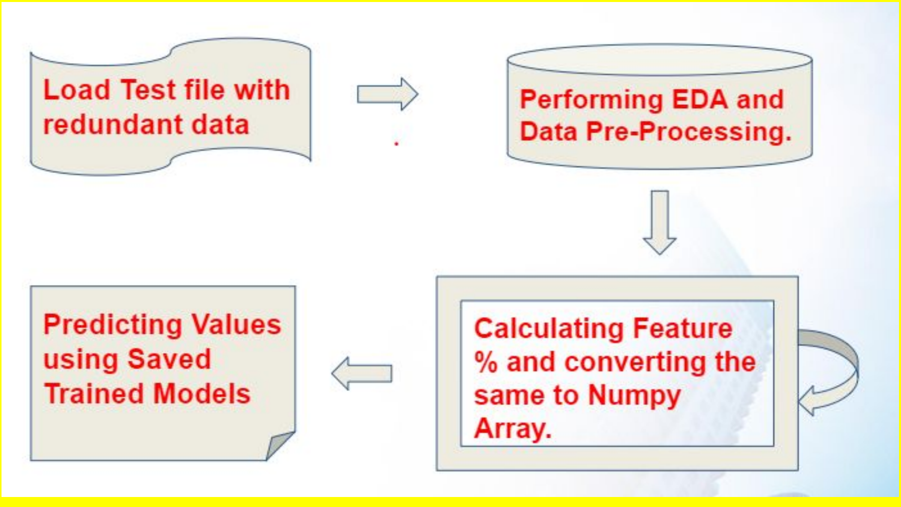
•The sponsor of the project also wanted to us to identify, which metrics are most important for data analysis and which are the ones that are being duplicated the most.

•The requirement of the client here is to identify different metrics of duplication through a basket analysis, this will indeed help them optimize the process of Data Acquisition and data-entry.

•3 segments were created for the data:

- ❏ 0%-50% Match: Basket 1.
- ❏ 50%-85% Match: Basket 3.
- ❏ 85%-100% Match: Basket 4.

Basket 1 being the least duplicated data and Basket 3 being the highest duplicated data, i.e. this basket has entries which have the highest match percentage of duplication or the highest similarity index ratio score.



Load Test file with redundant data

Performing EDA and Data Pre-Processing.

Calculating Feature % and converting the same to Numpy Array.

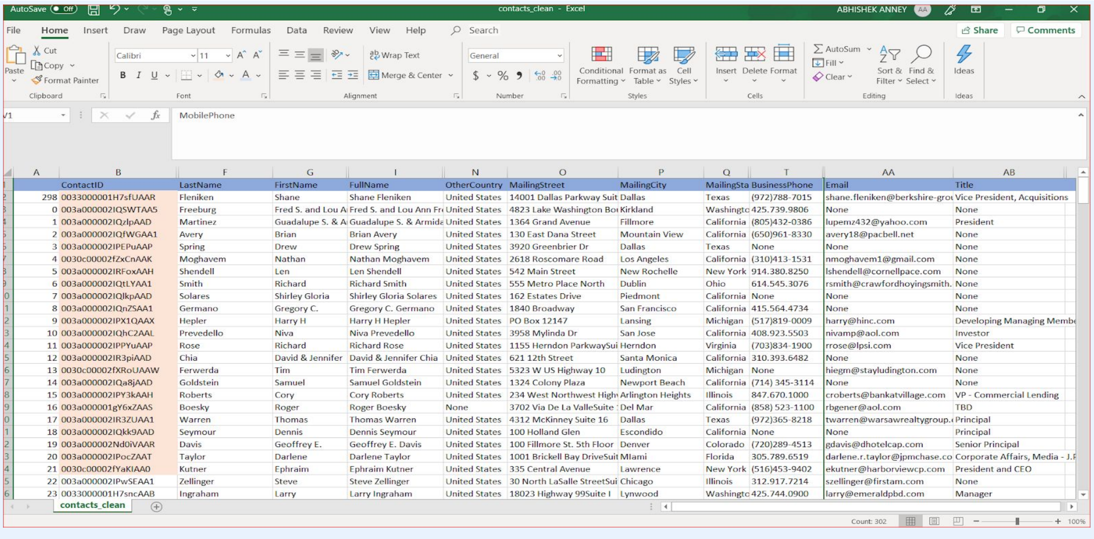
Predicting Values using Saved Trained Models

SEGREGATION OF DATA:

Below data has been received after processing 5000 Rows of data that was provided. Complete processing of 200K Rows will take high-performance machines with optimized specifications. The same has already been communicated to the client.

The final O/P after the execution of the models will be 2 excel files as below:

Cleaned Data:

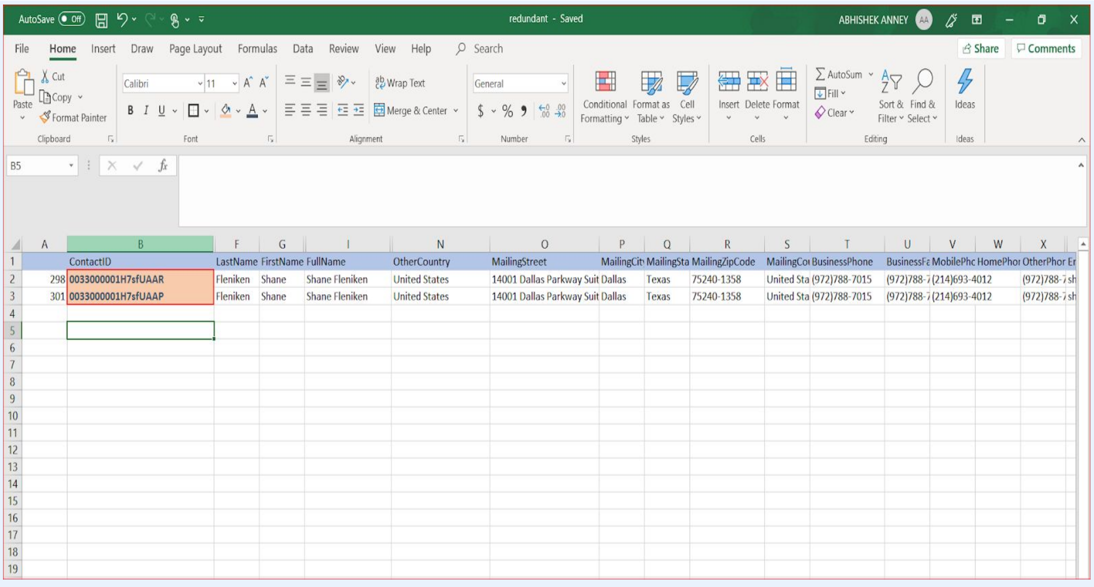


A	ContactID	LastName	FirstName	FullName	OtherCountry	MailingStreet	MailingCity	MailingSta	BusinessPhone	Email	Title
298	0033000001H7sfUAAR	Fleniken	Shane	Shane Fleniken	United States	14001 Dallas Parkway Suit	Dallas	Texas	(972)788-7015	shane.fleniken@berkshire-grou	Vice President, Acquisitions
0	003a000002IQSWTAA5	Freeburg	Fred S. and Lou A	Fred S. and Lou Ann Fr	United States	4823 Lake Washington Bo	Kirkland	Washingtc	425.739.9806	None	None
1	003a000002IQzlpAAD	Martinez	Guadalupe S. & A	Guadalupe S. & Armida	United States	1364 Grand Avenue	Fillmore	California	(805)432-0386	lupemz432@yahoo.com	President
2	003a000002IQfWGAA1	Avery	Brian	Brian Avery	United States	130 East Dana Street	Mountain View	California	(650)961-8330	avery18@pacbell.net	None
3	003a000002IPEPuAAP	Spring	Drew	Drew Spring	United States	3920 Greenbrier Dr	Dallas	Texas	None	None	None
4	0030c00002fZxCnAAK	Moghavem	Nathan	Nathan Moghavem	United States	2618 Roscomare Road	Los Angeles	California	(310)413-1531	nmoghavem1@gmail.com	None
5	003a000002IRFoxAAH	Shendell	Len	Len Shendell	United States	542 Main Street	New Rochelle	New York	914.380.8250	lshendell@cornellpace.com	None
6	003a000002IQtLYAA1	Smith	Richard	Richard Smith	United States	555 Metro Place North	Dublin	Ohio	614.545.3076	rsmith@crawfordhoyingsmith.	None
7	003a000002IQlkpAAD	Solares	Shirley Gloria	Shirley Gloria Solares	United States	162 Estates Drive	Piedmont	California	None	None	None
8	003a000002IQnZSAA1	Germano	Gregory C.	Gregory C. Germano	United States	1840 Broadway	San Francisco	California	415.564.4734	None	None
9	003a000002IPX1QAAX	Hepler	Harry H	Harry H Hepler	United States	PO Box 12147	Lansing	Michigan	(517)819-0009	harry@hinc.com	Developing Managing Membe
10	003a000002IQhC2AAL	Prevedello	Niva	Niva Prevedello	United States	3958 Mylinda Dr	San Jose	California	408.923.5503	nivamp@aol.com	Investor
11	003a000002IPPYuAAP	Rose	Richard	Richard Rose	United States	1155 Herndon ParkwaySui	Herndon	Virginia	(703)834-1900	rrose@lpsi.com	Vice President
12	003a000002IR3piAAD	Chia	David & Jennifer	David & Jennifer Chia	United States	621 12th Street	Santa Monica	California	310.393.6482	None	None
13	0030c00002fXRoUAAW	Ferwerda	Tim	Tim Ferwerda	United States	5323 W US Highway 10	Ludington	Michigan	None	hiegm@stayludington.com	None
14	003a000002IQa8jAAD	Goldstein	Samuel	Samuel Goldstein	United States	1324 Colony Plaza	Newport Beach	California	(714) 345-3114	None	None
15	003a000002IPY3kAAH	Roberts	Cory	Cory Roberts	United States	234 West Northwest High	Arlington Heights	Illinois	847.670.1000	croberts@bankatvillage.com	VP - Commercial Lending
16	003a000001gY6xZAAS	Boesky	Roger	Roger Boesky	None	3702 Via De La ValleSuite	Del Mar	California	(858) 523-1100	rbgener@aol.com	TBD
17	003a000002IR3ZUAA1	Warren	Thomas	Thomas Warren	United States	4312 McKinney Suite 16	Dallas	Texas	(972)365-8218	twarren@warsawrealtygroup.c	Principal
18	003a000002IQkk9AAD	Seymour	Dennis	Dennis Seymour	United States	100 Holland Glen	Escondido	California	None	None	Principal
19	003a000002Nd0iVAAR	Davis	Geoffrey E.	Geoffrey E. Davis	United States	100 Fillmore St. 5th Floor	Denver	Colorado	(720)289-4513	gdavis@dhotelcap.com	Senior Principal
20	003a000002IPocZAAT	Taylor	Darlene	Darlene Taylor	United States	1001 Brickell Bay DriveSuit	MIami	Florida	305.789.6519	darlene.r.taylor@jpmchase.co	Corporate Affairs, Media - J.F
21	0030c00002fYaKIAA0	Kutner	Ephraim	Ephraim Kutner	United States	335 Central Avenue	Lawrence	New York	(516)453-9402	ekutner@harborviewcp.com	President and CEO
22	003a000002IPwSEAA1	Zellinger	Steve	Steve Zellinger	United States	30 North LaSalle StreetSui	Chicago	Illinois	312.917.7214	szellinger@firstam.com	None
23	0033000001H7sncAAB	Ingraham	Larry	Larry Ingraham	United States	18023 Highway 99Suite I	Lynwood	Washingtc	425.744.0900	larry@emeraldpbd.com	Manager

contacts_clean

Count: 302

Redundant Data Output:



A	ContactID	LastName	FirstName	FullName	OtherCountry	MailingStreet	MailingCit	MailingSta	MailingZipCode	MailingCo	BusinessPhone	BusinessFa	MobilePhc	HomePhor	OtherPhor	Er
298	0033000001H7sfUAAR	Fleniken	Shane	Shane Fleniken	United States	14001 Dallas Parkway Suit	Dallas	Texas	75240-1358	United Sta	(972)788-7015	(972)788-7	(214)693-4012		(972)788-7	sh
301	0033000001H7sfUAAP	Fleniken	Shane	Shane Fleniken	United States	14001 Dallas Parkway Suit	Dallas	Texas	75240-1358	United Sta	(972)788-7015	(972)788-7	(214)693-4012		(972)788-7	sh

NEXT STEPS AND CONCLUSION:

➔ We discussed with the sponsor/academic advisor for the use of multiple websites which are available for use and designed by companies who perform the requirements of the project on a bigger scale. The sponsor advised us not to use any assistance or support from external websites or portals/API services.

➔ Choosing the right variables and the metrics based on the acquired domain knowledge data through the data dictionary provided by the client.

➔ Since the data-set is huge, we processed the data in multiple batches with high run-time. We will inform the sponsor that it is required to have a computer with high configuration specifications, in order to pre-process the data of a month(Around 2M rows) together.

➔ Address verification using Google Map API Integration.

➔ Email Verification using Zero Bounce API.

➔ Below technologies have been selected for the project:

- Python (Programming Language)
- Pycharm as IDE.
- Pandas (Data Preprocessing)
- Fuzzywuzzy (Percentage based Matching)
- Scikit-learn (Machine Learning Algorithms)
- Numpy (Numerical Calculations)

➔ Project design and implementation went smoothly. The team coordinated and supported each other to overcome the road blockers and challenges.