# Automated Sanction Checker using ChatBot

**Akilesh Balaji**

2993205

Submitted in partial fulfillment for the degree of

Master of Science in Big Data Management and Analytics

Griffith College Dublin

June, 2020

Under the supervision of Supervisor's Name

Prof. Bilal Yousuf

**Disclaimer**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Master of Science in Applied Digital Media at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

**Signed: __Akilesh Balaji_____ _Date: 12-06-2020_____**

**Acknowledgements:**

With the deepest gratitude and humility I would like to thank my supervisor Prof.Bilal Yousuf, who's constant guidance and nurturing, helped me navigate myself through the project and successfully complete it.

Nothing would have been possible without the platform that was offered to me by Griffith College Dublin and it's administrative faculties whose timely delivery of information seemed to be crucial in finishing the job on the right pace.

My friends and family cannot be appreciated enough for the unconditional support and encouragment that they provided at times it was badly needed in the quest to stay focused to finish the job. Deeply Grateful and Humble.

**Table of Contents**

**List of Tables**

# Abstract

*Artificial Intelligence in recent times, has evolved into one of the essential cogs in the quest to counter Financial Crimes, marking an inherent significance in banking sector. Such prominence has been enhanced further by the facilitation of new metrics like Know Your Customer Data and Anti-Money Laundering Algorithms, that help in tracking malicious transactions and future vicious risks. These metrics are implemented to perform high-profile client verification that improves the due diligence process in the organization. Moreover, Machine Learning augmented with Artificial Intelligence seek to boost the prospects by providing automation of the system, which in turn ought to reduce the processing time to a drastic level. So, in-order to attain sustainable prosperity in fight the against Financial Crimes, it is ideal to deploy Automated Machine Learning Algorithms that are in compliance with the AML regulations. Thus, keeping the following statement as the focal point, this paper has been attempted to dissect on how Artificial Intelligence retaliates to Money Laundering and also to develop an Interactive Automated Responsive application aided by decisive Machine Learning Algorithms.*

# Chapter 1. Introduction

## 1.1 PREFACE:

For decades now, Financial Institutions have tried to navigate through a highly compliance landscape, facing the plea for increased transparency from customers and regulators, and cope with financial malfeasance while also minimising the risk of conducts. [AML Intro]. The inadvertent use of the banking system for money laundering activities is a key challenge facing the financial services industry. In response, regulatory authorities have introduced anti-money laundering (AML) regulations to detect and prevent such activities. The seismic raise of Artificial Intelligence through the years has given the aid to help solve such issues. FIs seem to deploy AI to dissect large chunks of data, to separate phony alerts out and identify high profile criminal conduct. It has been able to track patterns and connection that can be morbidly difficult to be picked up either rule-based monitoring or the human eye.[intro AML] It had made considerably grounds in the quest to solve issues in the field of AML.

## 1.2 GOALS AND MOTIVATION:

During the course of the recent pandemic crisis, independent and small businesses have found it difficult to make ends meet. The pain and uncertainty [Intro1] of such pandemics can make the aid's offered by any government seem inadequate. Another concern being that, even though certain governments seem to offer a structured aid scheme, yet only the bigger companies were benefitted from it leaving the independent business owners to suffer. This happened due to fact that the government outsourced the offer relief funds through private independent banks, thus causing a bias in distribution of the money. In-order, to solve such similar consequences in the future, Virtual Banking has to be made an astounding reality. The apt solution for this, is to come up with a fool-proof, rule- based algorithm that provides unbiased screening of customers.

This idea seems to go on to build the foundation for this entire project, in which an application of an Automated Interactive Chatbot supported by a reliable Optical Character Recognition tool, is built to provide unbiased screening of Customers, based on the Financial Sanction List.

## 1.3 RESEARCH QUESTIONS:

- **With an increased effort made to deploy AI in countering AML, can Machine Learning help counteract to complex financial misconducts?**
- **With multiple organizations adopting AI to build sanction screening models, can Financial Institutions be regulated to provide reliable Financial Sanction Lists in the future?**
- **Can the text pattern of Identification Documents be dissected efficiently using Natural Language Processing?**
- **Can Machine Learning help to improve the accuracy of the Sanction Checker Model?**

## 1.4 OVERVIEW OF APPROACH:

The customer requesting for a loan is asked for a document that serves the purpose of providing the bank with KYC data (Know Your Customer). The KYC data is streamed using the centralized spark RDD for the purpose of identification of the customer. The KYC can be obtained using any legal document owned by the customer (i.e. Passport etc).

Further OCR (Optical Character Recognition) is done to the document using python and based on the data from the document, the NLP process of Topic Modelling is done to segregate the customers. Then the corresponding data is stored in a Relational Database preferably Postgres SQL.

Then OFAC (Office of Foreign Asset Control) sanction list that is an open source data that consists of blacklisted personnel is retrieved and it is stored in the database. Then using the sanction list and the KYC the customer an AML algorithm is introduced and the risk factor score of the customer is calculated.

The following data is regressed with the risk factor score and using the manually generated mock data and with machine learning process of linear regression, the legitimacy of the risk score is validated.
Then finally using the risk factor score and the customer is provided with an automated narrative on whether he or she is eligible to apply for loan. The customer is interacted with a chatter bot. The narration is done using the Natural Language Generation.

## 1.5 OBJECTIVES:

The following are the various objectives that are to be achieved during the course of this project:

- **To build a chatbot that provides an interactive experience for the user**
- **To develop a prominent Optical Character Recognition tool that scans the identification documents.**
- **To formulate an Algorithm that provides High Level Screening of Customer based on KYC data**
- **To model various Machine Learning Algorithms that evaluates the efficiency of the application.**

## 1.6 STRUCTURE OF DOCUMENTATION:

The documentation is done to interpret the idea and the motivation behind developing the application. In Chapter II, various research papers have been gathered and carefully analysed to provide the background study to track the evolution of AML algorithms through the years. Critical Analysis of various Sanction Checker in the market is also done. In Chapter III, various technological commodities used to build the sanction checker is discussed. In Chapter IV, state of the art Application model design and Specification is discussed. In Chapter V, the process involved to deploy the Sanction checker is discussed systematically. In Chapter VI, various evaluation models and metrics used to validate the efficiency of the application is discussed. Finally, the future works related to the application is discussed, followed by the conclusion.

# Chapter 2. Background

## 2.1 LITERATURE REVIEW:

Banks and financial institutions are facing some serious AML challenges in recent times, that can be typically attributed to faulty mitigation approaches. In the current framework money laundering refers to as the process of shading the illicit origins of the wealth acquired from crime making it a predicate offence. The aim of AML is to make the predicate offence less profitable. [3]
The need for personal accountability for corporate acts has become common in the current banking regulatory environment. Firms that fail to prevent laundering tend to pay a heavy price in the form of declining revenues, customer dissatisfaction, huge penalties, loss of reputation, and fall in stock prices. [1]

Recent shortcomings of the banking framework had been laid bare by a stint of high-profile scandals that had the potency to shut down banks and slash the lender's share prices among various European countries. The lack of synergy between the banking bodies across EU has been identified as one of the key weaknesses exploited by the money launderer's [2]. In-order for countries to meet international standards to combat illicit drug trade, financing of terrorism and organized crimes, the corresponding countries must adopt various universal codes and regulations that make the AML systems. So far developing countries have found it less feasible to employ AML systems as an anti-corruption medium yet. On the other hand the costs of AML systems in developing countries have probably outweighed the benefits. In addressing such shortcomings of the AML system, Reuter and Truman state that" It is an article of faith to the authorities in industrial countries that all nations need to have effective AML regimes, but resources are scarce. The global threat posed by weaknesses in poor countries may be quite minor".[3]

This has led the Federal Financial Reserve Agency of Washington to instruct various banks to take innovative measures to negate such threat. The agencies said they would not penalize banks for initiating pilot programs, for one, nor would they penalize banks if those pilot programs ultimately prove unsuccessful. Supervisors also said they would not necessarily apply new or additional sanctions on banks that employ innovative strategies and through that innovation discover previously unknown threads of illicit financial transactions [4]

One of the more exciting benefits of looking forward to the potential of technology to detect previously unknown patterns, and this will deepen our understanding of financial crime and how to prevent our bank from running afoul of AML laws and regulations around the world," said Loretta Yuen, group general counsel of OCBC. Certain U.S. bankers have wondered if unsupervised machine learning could catch innocent people in a spot of bother. But Yuen does not see this as an issue because the technology does not look at just one red flag, but at a range of parameters, including products, customers and risks, to separate bad guys from good ones. At the bank, internal analysts will watch the software closely to ensure the technology is performing consistently and as intended in all situations.[5]

Various banks are using three popular software modules from QuantaVerse.[4]. Other vendors of AI software that can be used to detect money laundering and other financial crime include Thetaray, Merlon Intelligence, ZestFinance, Ayasdi, Quantexa, and IBM Watson. One module by QuantaVerse is called Pre-TMS(Transaction Monitoring System) Entity Resolution & Risk Scoring, which is intended to reduce false positive alerts by classifying the risk of each transacting party. This software works to find missing information about people and companies and clean up information. A transaction conducted on behalf of a company whose owner was indicted for financial crimes in a previous job would be labeled high risk. The second piece of software is called Alert Investigator. It helps human compliance officers investigate financial crime alerts. This can provide the investigator an automated financial crime report with natural language generation that includes a recommendation of whether to file a suspicious activity report or not or if further investigation is recommended. The third component is a false negative

identifier. This looks at all the transactions that were not alerted and analysing them to make sure nothing nefarious was missed. The bank's top goal is to improve efficiency in its investigative process as it grows. Its second goal is to catch every instance of financial crime in its organization and prevent money launderers, drug traffickers and human traffickers from using its rails.[6]

Money laundering is a low-frequency event, but banks can pay a high price for missing an incident. To detect money laundering, banks deploy monitoring systems to alert them of atypical transactions. Based on certain criteria, a financial-investigations unit then attempts to identify likely instances of money laundering from among the alerts, filing suspicious-activity reports with appropriate authorities as needed.
But anti-money laundering (AML) operations are often hampered by high levels of false positives--much higher than you would expect. Here's why: A very effective transaction-monitoring system might be 95 percent specific for suspicious activity and 95 percent accurate in detecting it. This means that the control falsely detects suspicious activity in 5 percent of normal cases while flagging 5 percent of all activity as not conforming to the established criteria. In those cases, further work will be needed to determine whether they are legitimate or suspicious. If, after all, 0.1 percent of transactions truly meet the criteria for suspicious activity (1 in 1,000 among the 50 in 1,000 falsely flagged), then this particular control will have produced a false-positive rate of more than 98 percent.. Fewer than 2 percent of alerts will correspond to activity that upon further examination qualifies as suspicious. [7]

During the course of developing an AML algorithm, there are two concerns that regulators have to address. First, consider how bias can enter into the anti-money-laundering reports that banks file with the Financial Crimes Enforcement Network. Structural barriers including linguistic or socioeconomic hurdles can inadvertently create bias within data. Another concern about regulators' reaction is around "explainability." Regulators say that banks should not make any decisions in a black box, but always be able to spell out the reasons for each decision. Some institutions fear that AI makes it harder to do so, but vendors insist that's not the case. Explainable artificial intelligence and AI bias continue to be top of mind for both banks and

examiners," said Nikhil Aggarwal, director at IBM's Promontory Financial Group.". Another fear that was addressed directly in the regulators' statement: If a new technology uncovers criminal activity that's been going on for years undetected, the bank will get in trouble for which the Financial agencies say they won't judge banks' old systems against the new. A last concern some financial executives have expressed about using AI to detect money laundering is that innocent people could get caught up in a dragnet. But most of the AI-based solutions present their findings to humans who review the anomalies and the reasons they were flagged.[6]

Not just based on banking patterns, but there are start-ups that use AI to track the PEP(Polically Exposed Person) relations.Take for example , Mr.Konrad Alt, a former regulator and chief operating officer of Promontory Financial Group, and artificial intelligence expert Bradford Cross who are launching a startup called Merlon Intelligence that plans to expose the PEP relations by improving upon identity verification by mapping connections between people inside and outside of the United States and monitoring their relationships. It monitors the movement of money for anything that might look like money laundering or other kinds of foul play. It regulates the negative news that banks are required to screen — literally scanning news mentions of customers or potential customers involved in somewhat wrongdoing misconduct.[6]

Although such technology tools might not completely eradicate money laundering, they will bring it under control to a considerably extent, and financial institutions should proactively look at adopting these sooner than later. [10]

## 2.2 RELATED WORKS:

## Case Study: Based on Organizations

### 2.2.1 Accenture

Various Software Enterprises have been developing their own version of the AML System and Algorithms. Accenture for example have been developing an application called the AML Alert Triage. From their perspective they think of ML to be a sub-field of AI where computers are being able to learn new data without being

programmed explicitly. The possible application seems to be vast among the activity if Suspicious Monitoring and Transaction Monitoring. [11]

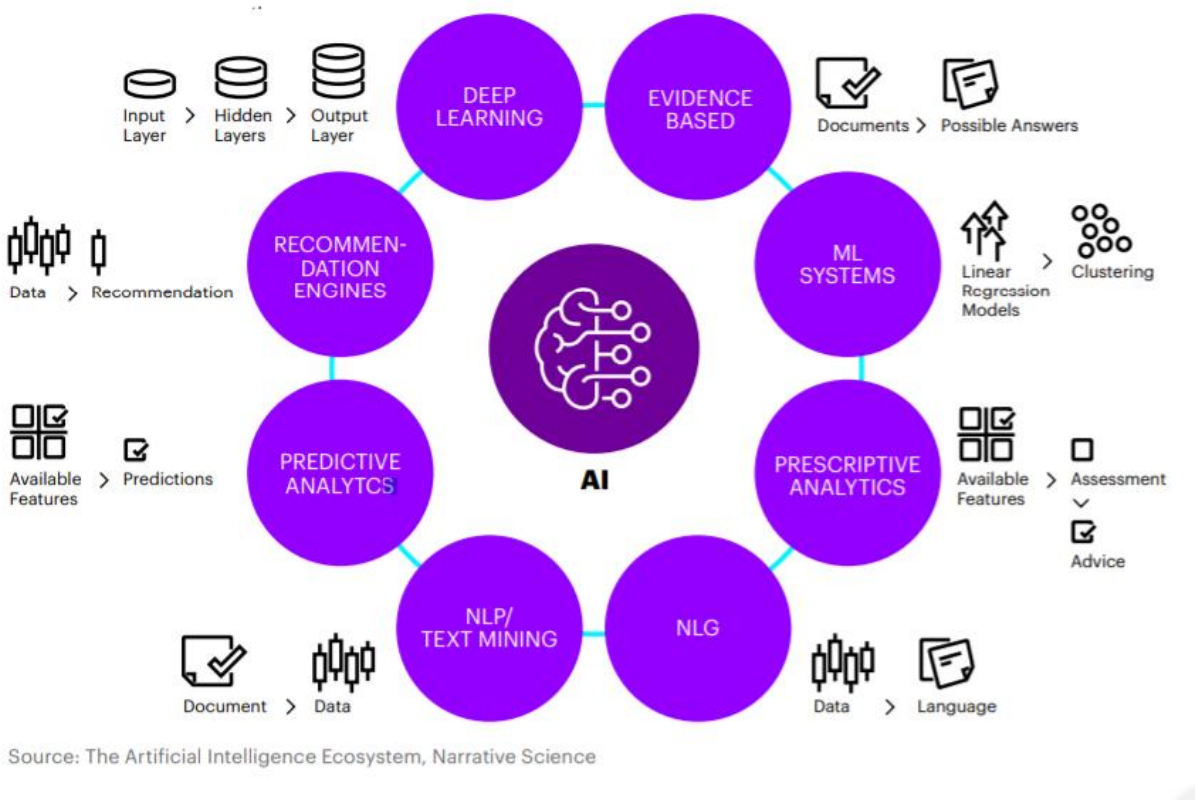



Fig 2.1

The common obstacles in Transaction Monitoring is the not so feasibility in the cost for prompting the triage to alert. With increase in customers there seems to be a huge increase in Volume of alerts. This in turn forces organization to have multiple personnels to keep track of the alerts consistently. With Automation such large volumes of Alert can be suppressed and can be made to hibernate at times. Machine Learning can teach computers to sense suspicious threats and segregate alerts based on risk-based approach.

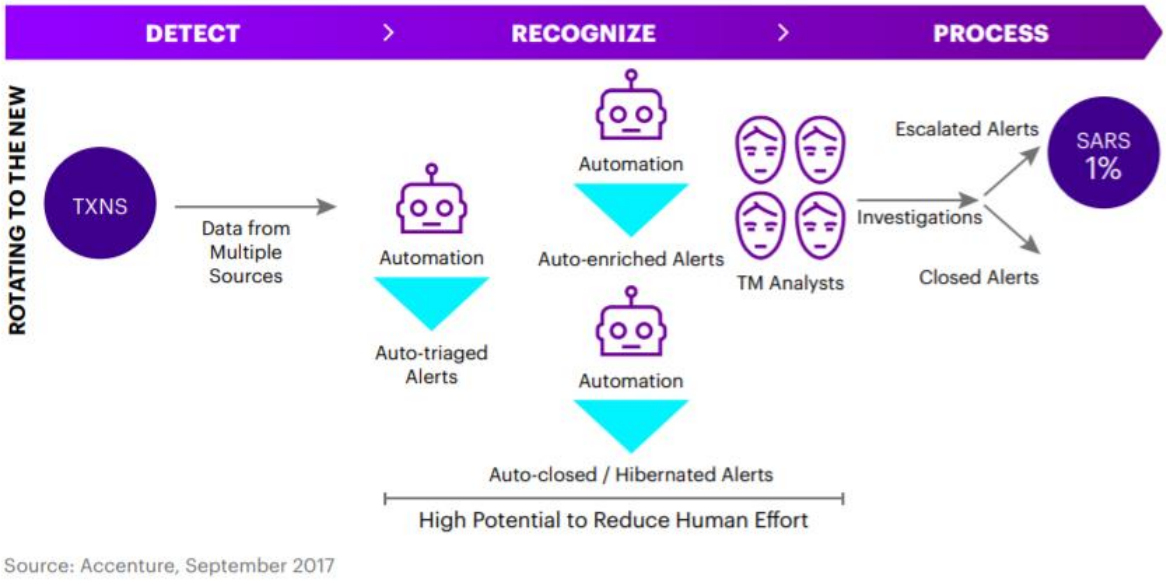



Fig 2.2

By Using Artificial Intelligence among the entire Transaction Monitoring, alerts can be automated. The customer KYC is obtained and using Natural Language Processing the data is processed and narratives for Suspicious Activity Report is Generated using Natural Language Generation. This can reduce dependencies of human operations. [8]

### 2.2.2 Bobsguide:

Next up according to bobsguide, AI is believed to be the future of AML systems. For each alerted case, the estimation for risk of money laundering is done used by various powerful algorithms such as Random Forests, Gradient-Boosted methods, (XGBoost, LightGBM, CatGBM) and deep neural networks. New[9]

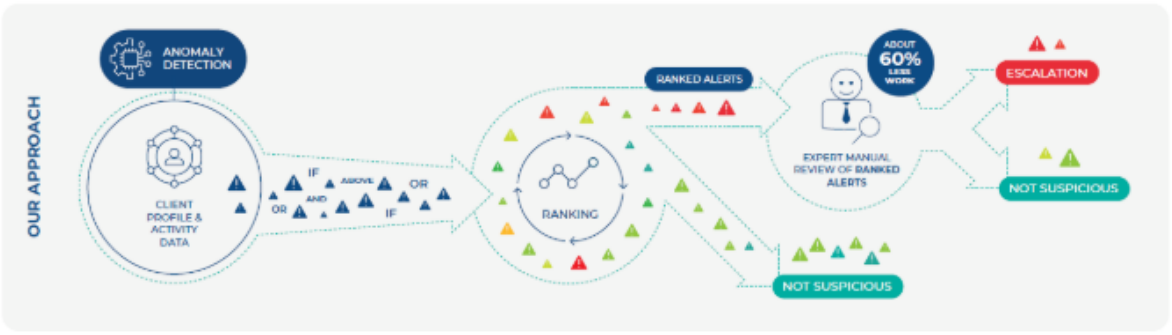



Fig 2.3

### 2.2.3 Ernst and Young:

In Ernst and Young, the framework used to detect KYC was given a whole new make-over. In this model they believe AI could enhance the breadth, scale and frequency to Know Your Customer Review that in-turn enhances the screening and monitoring Analysis. Risk Models can be detected from a enriched set of inputs and they can produce outputs based on the customer's profile and cognitive behaviour. By coupling dynamic unsupervised learning along with skilled investigators, the model can used to provide quality results and improve training in the future.

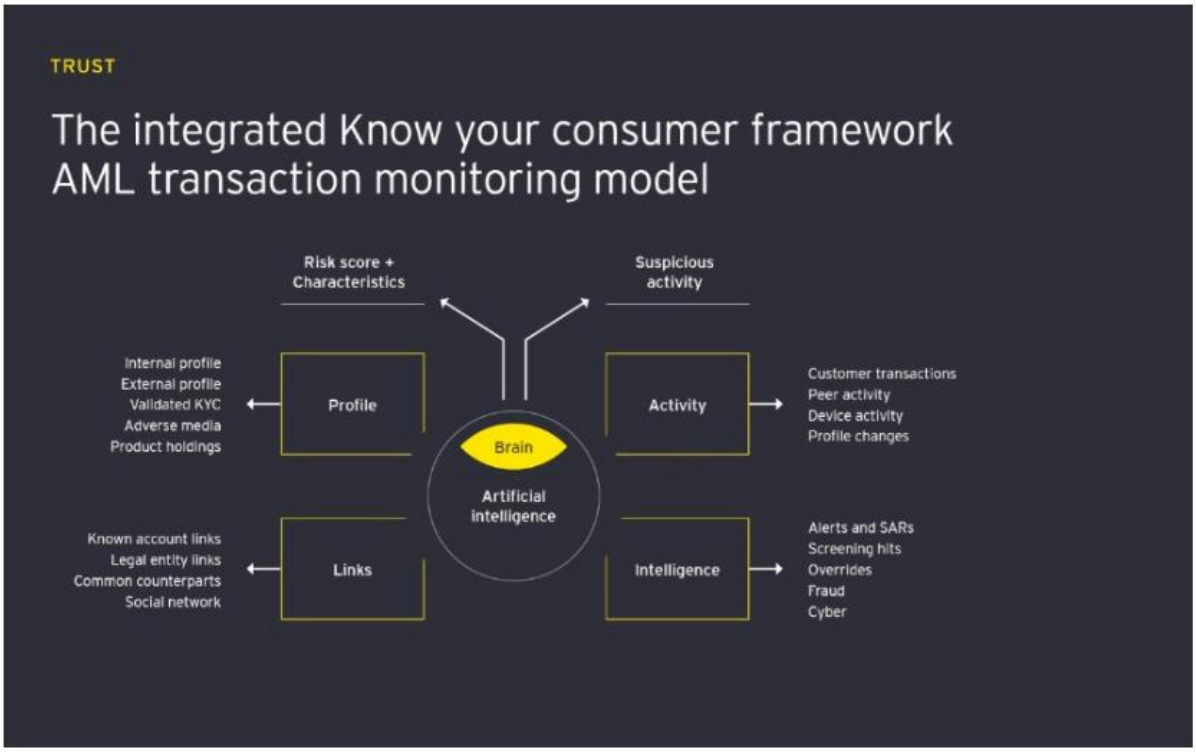



Fig 2.4

**2.2.4 Infosys:**

In Infosys even though the system seems to be adopting a traditional sanction screening workflow, the text matching algorithm that was adopted was the Fuzzy match. There are numerous sanction checkers that does not use Machine Learning as part of the checking process. In such cases a Machine Learning plugin approach is adopted to make the system compatible with the DB architecture of the company. [9]

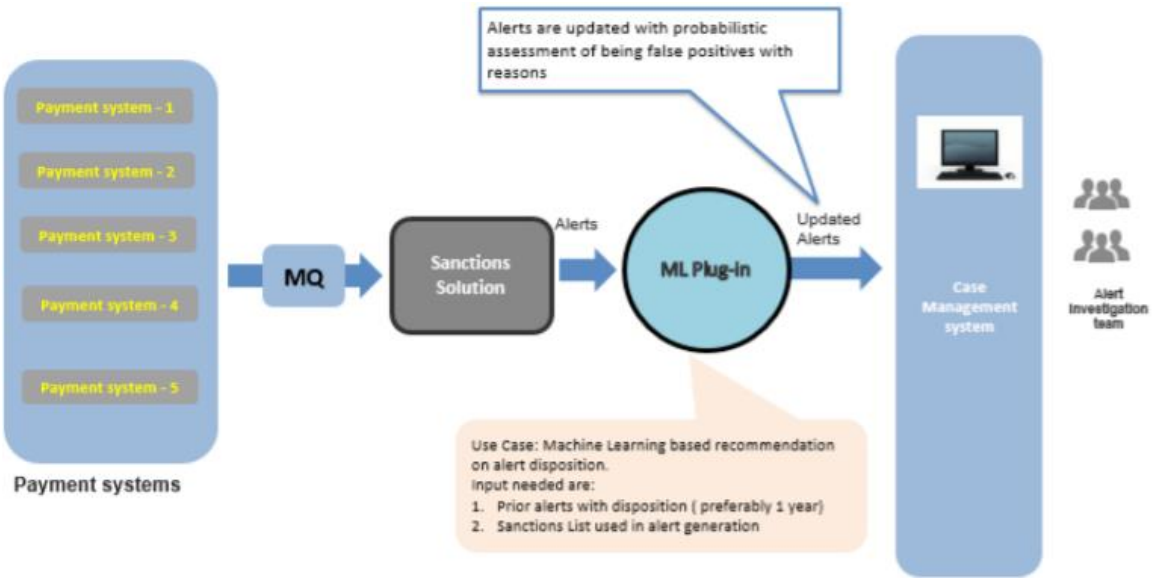



Fig 2.5

**2.2.5 IBM:**

IBM insists that to out-pace fraudsters, financial institutions and payment processors need a quicker and more agile approach to payment fraud detection. Instead of relying on predefined models, applications need the ability to quickly adapt to emerging fraud activities and implement rules to stop those fraud types. Not only should organizations be able to adjust their detection models, the models themselves should be inter-operable with any data science, ML, open source and AI technique—using any vendor. In addition, to eliminate fraud traveling from one area or channel to another undetected, aggregating transactional and non-transactional behavior from across various channels providers greater context and spots seemingly innocuous patterns that connect complex fraud schemes.[10]

**2.3 CRITICAL ANALYSIS:**

**Case Study: Based on Independent Products:**

**2.3.1 Name Scan:**

Name Scan is the company that provides financial sanction checkers that provide businesses, the service they require under AML regulation and help in the fight against money laundering and terrorism financing.

They use KYC data to analyse the customer data and try to reduce the risk of being associated with terrorist financial activities and money laundering. They also provide the users with multiple well known sanction lists, that allow the users to make precise screening of their customers.

**User Interface:**

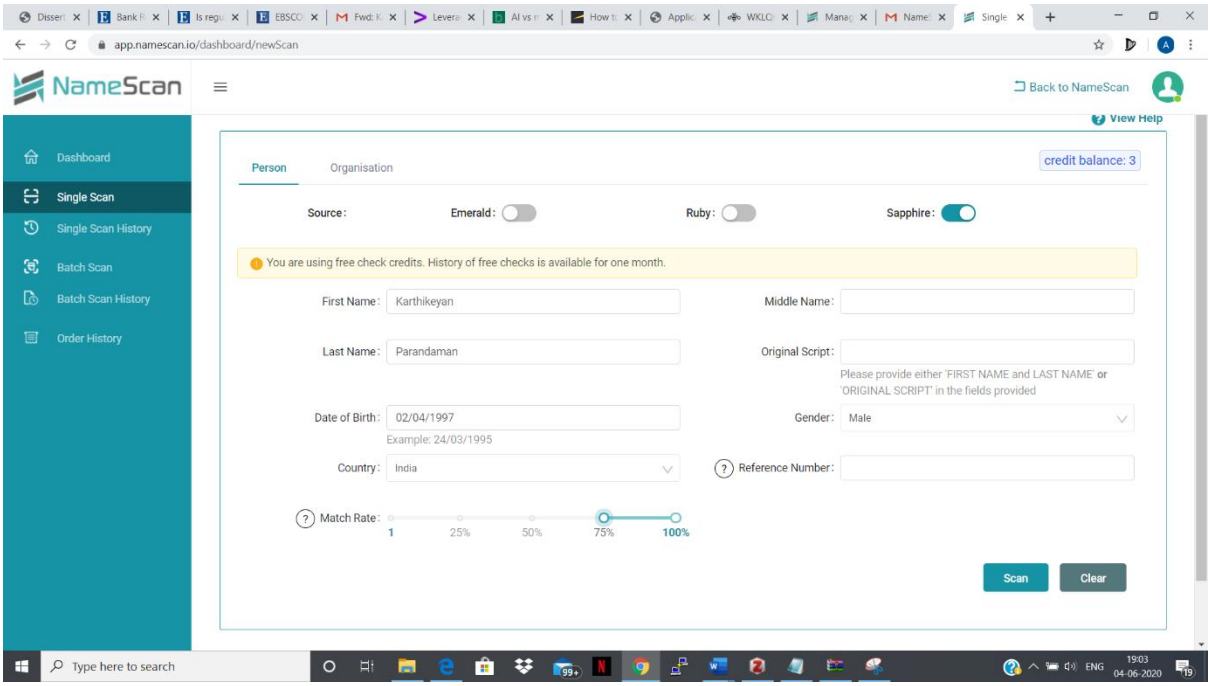



Fig 2.6

**2.3.2 Lexix Nexis:**

LexisNexis Risk Solutions is a leader in providing essential information that helps advance industry and society. It is used by both government and Commercial Organizations.

**2.3.3 Refinitiv:**

Refinitiv World-Check is a risk intelligence database which helps organizations across the world meet their regulatory obligations, make informed decisions and help prevent them from inadvertently being used to launder the proceeds of financial crime or association with corrupt business practices. It operates alongside Harlequins Data Analytics that help build and maintain the Sanction Checker.

**Comparison Table**:

| Name | NameScan | Refinitiv | LexixNexis | ChatterBOT |
|---|---|---|---|---|
| Sanction List | Yes | Yes | Yes | Yes |
| User Interface | UIx | UIx | UIx | ChatBOT |
| Autoamted Loan Sanction Screening | No | No | No | Yes |

Table 2.1

# Chapter 3. Methodology

## 3.1 OPTICAL CHARACTER RECOGNITION:

Optical character recognition (OCR) is a system that converts the input text into machine encoded format. OCR serves its users in converting the typewritten documents into digital form. This has made the retrieval of the required information easier as one doesn't have to go through the piles of documents and files to search the required information. Organizations are satisfying the needs of digital preservation of historic data, law documents, educational persistence etc. An OCR system depends mainly, on the extraction of features and discrimination / classification of these features (based on patterns. )[16]

In the current decade, researchers have worked on different machine learning approaches which include Support Vector Machine (SVM), Random Forests (RF), k Nearest Neighbor (kNN), Decision Tree (DT) etc. Researchers combined these machine learning techniques with image processing techniques to increase accuracy of optical character recognition system. Recently researchers has focused on developing techniques for the digitization of handwritten documents, primarily based on deep learning [20] approach.[17] This paradigm shift has been sparked due to adaption of cluster computing and GPUs and better performance by deep learning architectures, which includes Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) networks etc.

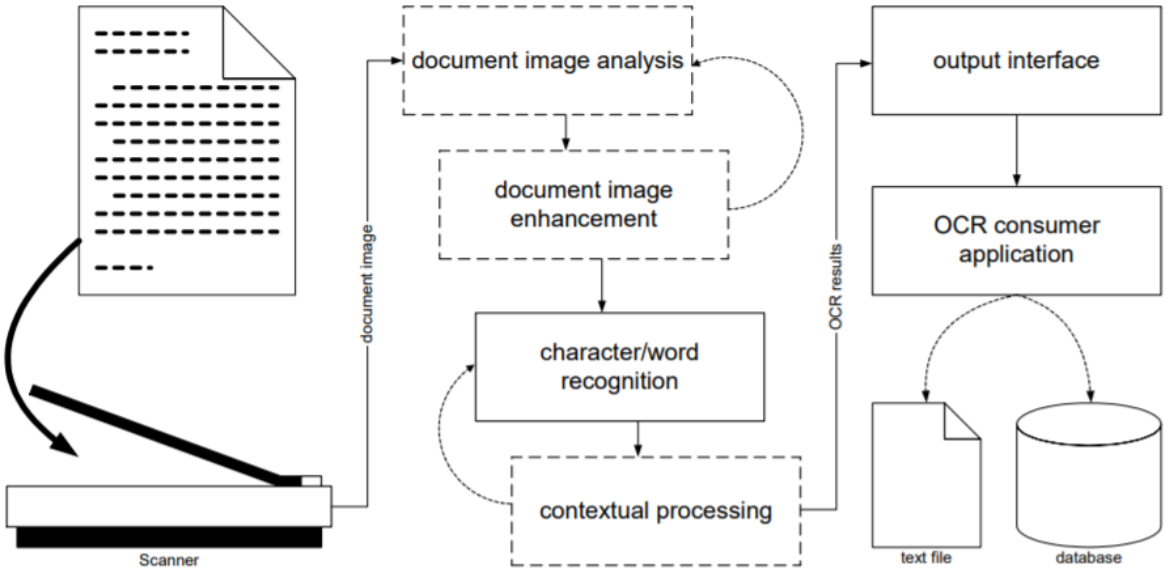



Figure 1: A typical OCR system

Fig 3.1

### 3.1.1Categories of OCR:

Commercial OCR systems can largely be grouped into two categories: task-specific readers and general purpose page readers.

A task-specific reader handles only specific document types. Some of the most common task-specific readers process standard forms, bank checks, credit card slips, etc. These readers usually utilize custom made image lift hardware that captures only a few predefined document regions. For example, a bank check reader may just scan the courtesy amount field and a postal OCR system may just scan the address block on a mail piece. Such systems emphasize high throughput rates and low error rates. Applications such as letter mail reading have common throughput rates of 12 letters per second with error rates less than 2%. The character recognizers in many task-specific readers are able to recognize both handwritten and machine-printed text.

General purpose page readers are designed to handle a broader range of documents such as business letters, technical writings and newspapers. A typical general purpose reader works by capturing an image of a document page, separating the page into text vs. non-text regions, applying OCR to the text regions and storing non-text regions separately from the output text. Most of the general purpose page readers can read machine written text but only a few can read hand-printed alphanumerics. High-end page readers have advanced recognition capabilities and high data throughput. Low-

end page readers usually are compatible with generic flat-bed scanners that are mostly used in an office environment with desk top computers, which are less demanding in terms of system accuracy or throughput. Some commercial OCR software is adaptive and allows users to fine-tune the optical recognition engine to customer's data for improved recognition accuracy.

### 3.1.2 Difficulty in OCR:

Character misclassifications stem from two main sources: poor quality recognition unit (item) images and inadequate discriminatory ability of the classifier. There are many factors that contribute to noisy, hard to recognize item imagery:

- Poor original document quality
- Noisy, low resolution, multi-generation image scanning
- Incorrect or insufficient image pre-processing
- Poor segmentation into recognition items

On the other hand, the character recognition method itself may lack a proper response on the given character (item) set, thus resulting in classification errors. This type of errors can be difficult to treat due to a limited training set or limited learning abilities of the classifier.

### 3.1.3 Scannable Documents:

Due to such a variety in document image formats, most commercial-off-the-shelf (COTS) OCR solutions work with more than a single image file format and can adapt to various spatial resolutions and pixel depths. Many COTS OCR packages come with a rich set of image processing utilities that analyze and transform input images to the format most suitable for the given OCR engine. ScanSoft Capture Development System 12 features a wide range of image and application format support, including BMP, GIF, TIF, PDF, HTML, Microsoft Office formats, XML, and Open eBook. ABBYY Fine Reader 7.0 can work with black-and-white, gray-scale and color images in various formats including BMP, PCX, DCX, JPEG, JPEG 2000, PNG, PDF, and TIFF (including multi-image, with the following compression methods: Unpacked, CCITT Group 3, CCITT Group 4 FAX(2D), CCITT Group4, PackBits, JPEG, ZIP)

Sakhr Automatic Reader 7.0 provides necessary functionality to read and write images in various formats including TIFF, PCX, BMP, WPG, and DCX. Most consumer-level OCR programs work with bi-level imagery and typically expect a black text on a white background. Some can import gray-scale images and internally convert them to black-and-white, sometimes using adaptive thresholding. Only rare OCR engines can directly work with gray-scale or color images taking advantage of multi-bit pixel information.

## 3.3 CHATBOT:

A chatbot is an application that provides interact responses to users in their natural language. They are also named in terms such as machine conversation system, virtual agent, dialogue system, and chatterbot. The chatbot system carries out its operations in the Natural Language Understanding (NLU) engine. Chatbot can be a text based, spoken or can be a non-flask communication. Chatbot can work both on PCs and smart phones, but mostly works on the internet.

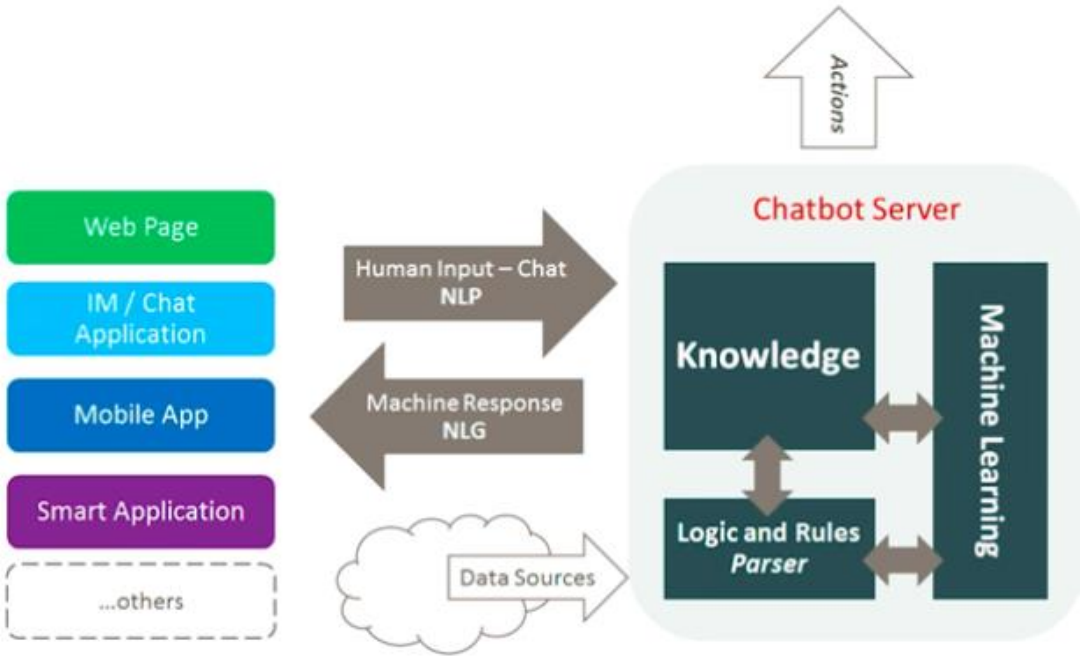



Fig 3.2

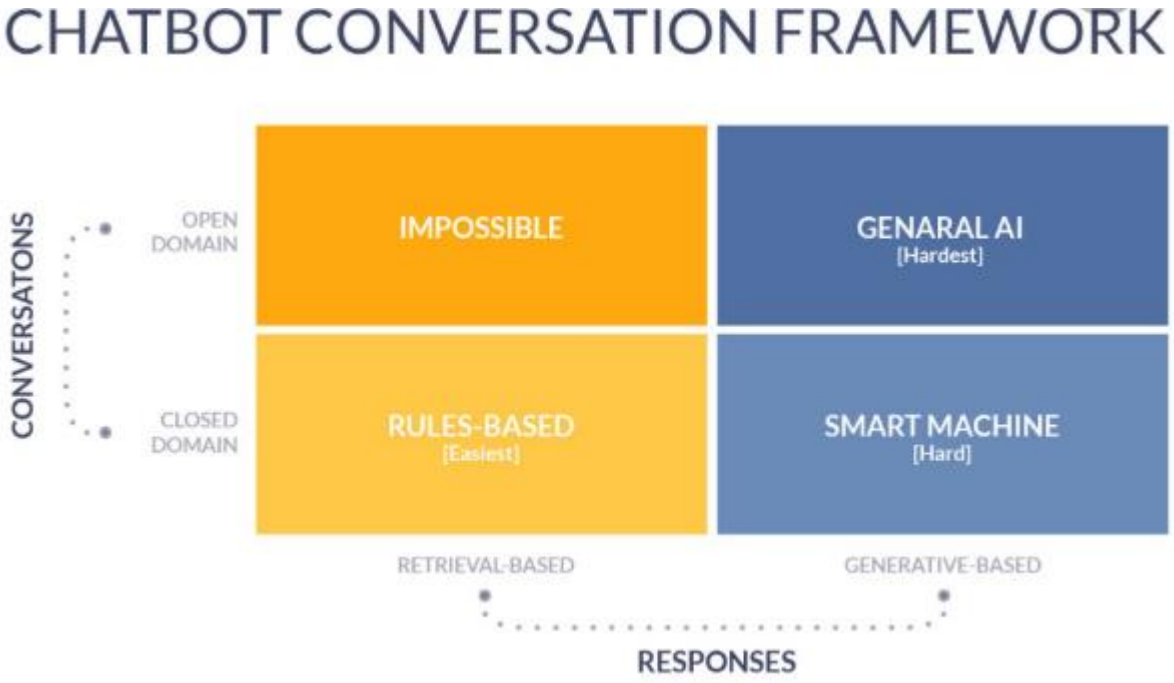



Fig 3.3

### 3.3.1 Need for Chatbot:

Human–computer interaction (HCI) mediates the redistribution of cognitive tasks between humans and machines. This technology, started in the 1960s, initially aimed to determine whether chatbot systems could make users into believing that they were real humans. The idea of conversing to a software seemed bizarre in the past. Now, with technological advancements our day to day life requires the help of chatbots. Some companies today have found that smartphone users dislike the idea of installing too many apps. For this reason, these companies are developing chatbots to make it more convenient for the phone users to find answers to their questions. By the use of chatbots with a user interface, people have the possibility of heading to a specific web site to ask questions and receive answers that are satisfactory.[21]
Social constructivism theory states that social interaction plays an important role in the development of cognition, with learning being manifest in the intellectual aptitude, cognitive strategies, motor skills, and dispositions people develop while working toward a goal within a community of others. A chatbot has the potential to be used in social contexts, since it has the ability to mimic human interaction. It therefore has

capability to promote social interaction between people and between the chatbot itself and individuals; they are socially and interactively oriented.

### 3.3.2 Selection of Chatbot Platform:

The selection of platform is depends on the needs of the chatbot and its developer. It is dependent on whether chatbot will be goal oriented, conversational or goal oriented with conversational abilities in chatbot?
Chatbot platforms can be divided into three major categories.

**A. Non programming chatbots:**

The type of chatbot is not technically oriented platforms. Codes for developing this platform are fairly simple and do not require much domain expertise.

**B. Conversational -Oriented- chatbots:**

In such type of chatbots, the users have the ability to interact with bot. These platforms require expertise on specification languages such as AIML (Artificial Intelligence Markup Language) which is used to model the user interactions.

**C. Chatbots by tech giants:**

Tech giants such as Google develop Api.ai, Facebook develops Wit.ai, Microsoft develops LUIS, Amazon develops Lex and IBM develops Watson.

Api.ai chatbot features:

• Using Intents and Contexts, Api.ai can generate very large and complex systems.
• Chatbot can handle the code proactively and decreases server-side coding.
• APi.ai can offer one-click integration with the several platforms such as Twitter, Facebook.

Wit.ai chatbot feature:

This type of chatbot offer lot of key advantages which are as follows.
• The story concept is very useful and powerful.

• Branches lead to better control the conversation also conditions on the actions.

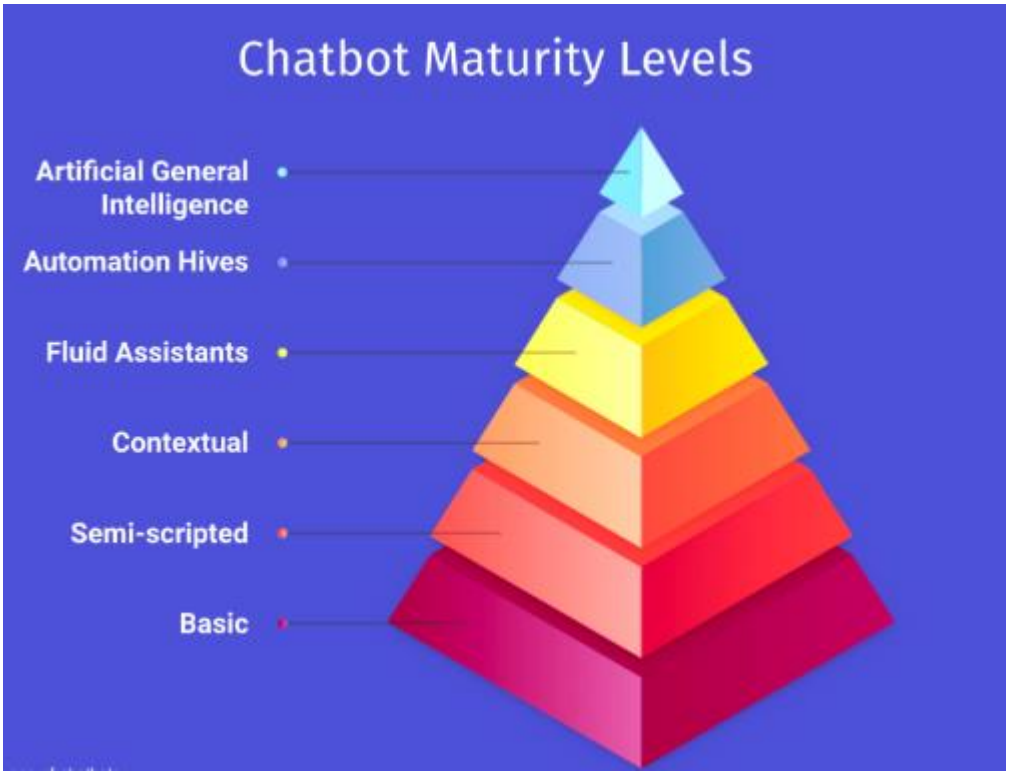



Fig 3.4

### 3.3.3Challenges associated with programming chatbot:

**A. Natural language processing:**

The fundamental challenge of the chatbot is to handle NLP issue by mastering their syntax. If we ask them that "what's the weather?" you will get an answer but what if we ask "Could you check the weather?" you might not get the proper answer.

**B. Machine learning:**

Machine Learning is another aspect of the Chatbot design and development. Our computer systems should be able to learn the correct response, which can be achieved with efficient programming using AI concepts.

## 3.4 GOOGLE BUCKET:

Google Cloud Platform (GCP) is one of the most important and growing in the cloud market. It provides developers to develop simple to complex programs. GCP offers hosting services on the same supporting infrastructure that Google uses internally for end-user products like Google Search and YouTube. This outstanding reliability results in GCP being adopted by eminent organizations such as Airbus, Coca-Cola, HTC, Spotify, etc. In addition, the number of GCP partners has also increased substantially, most notably Equinix, Intel and Red Hat.

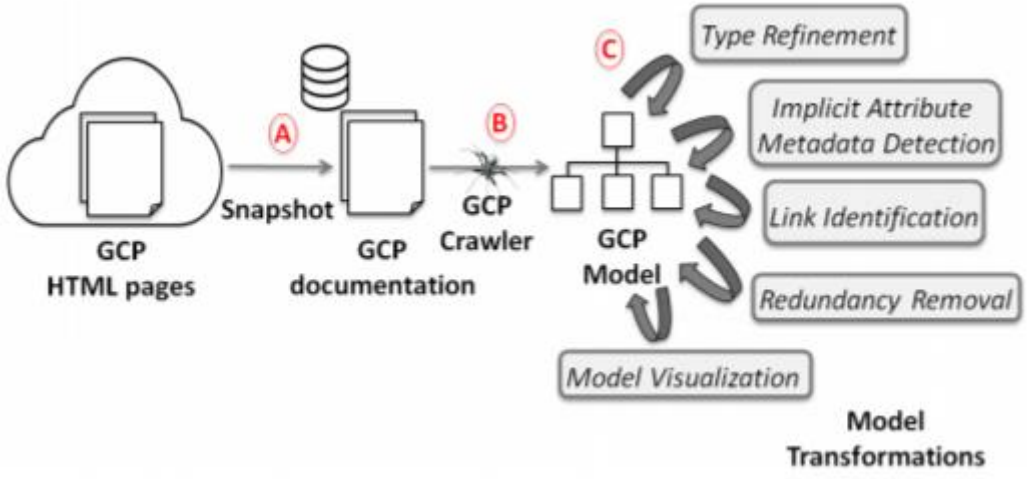



Fig 3.5

Google cloud storage buckets offers flexible, reliable and scalable storage options for a computer engine. They can share instant data across multiple instances and zone. Performance of cloud storage buckets depends on location of bucket and storage class. All storage classes are essentially the same in throughput but tend to differ in their availability, minimum storage durations and pricing.[23]

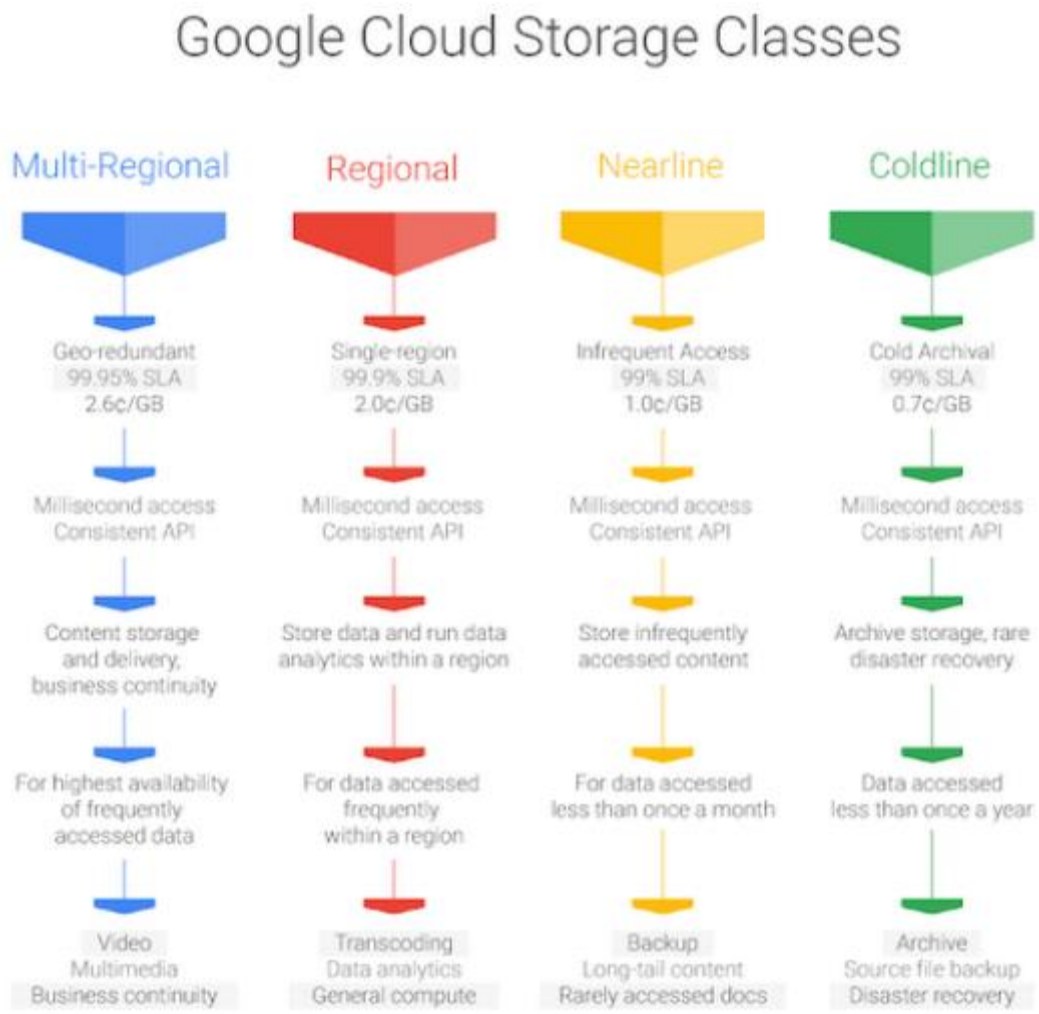



Fig 3.6

## 3.5 JUPYTER NOTEBOOK:

Jupyter Notebook is a cross-platform computational notebook interface implemented in Python with a web-based frontend. It launched by the name IPython Notebook and only supported Python-based notebooks. At the present, it has been extended to allow using languages other than Python. Jupyter *kernel*: a language-specific backend provides support for each language by receiving input from Jupyter Notebook, processes it using the target language, and returns the results to Jupyter. Communication by Jupyter with kernels is enabled through a language-agnostic protocol, which allows implementing kernels in languages other than Python. The

Jupyter Notebook web interface is extensive which includes using JavaScript-based plugins. This capability was used to implement syntax highlighting for B.

The Jupyter Notebook can be assessed using modern web browser making it easier for the user to run the same interface locally like a desktop application, or running on a remote server. The only software the user needs locally is a web browser; so, for instance, a teacher can set up the software on a server and easily give students access. The notebook files created are a simple, documented JSON format, with the extension '.ipynb'.

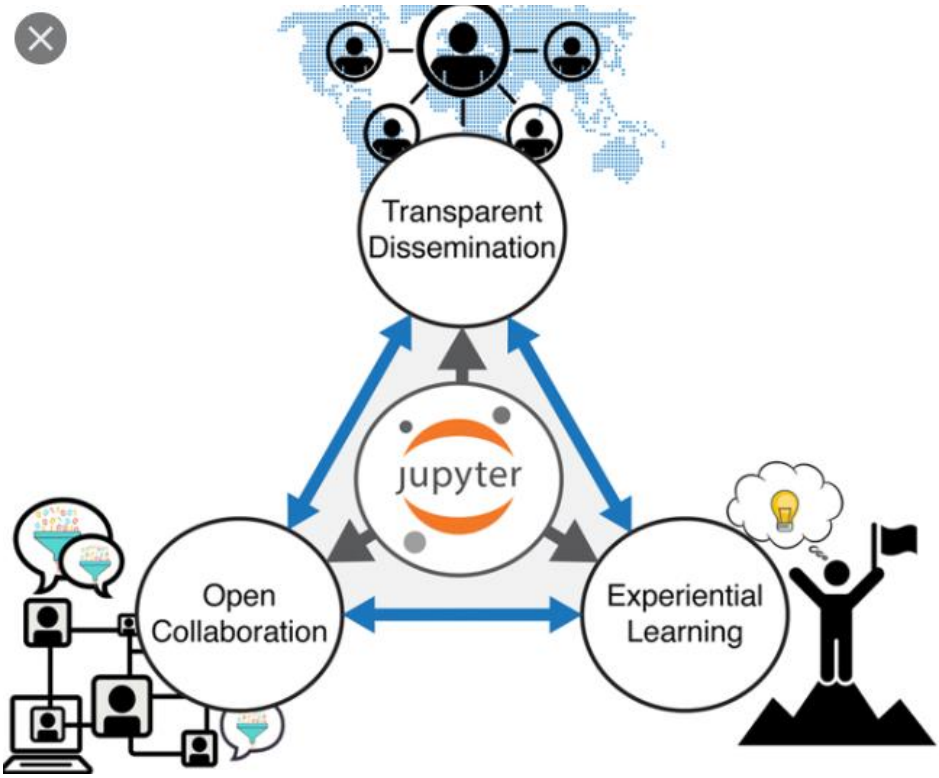



Fig 3.7

## 3.6 PYTHON-FLASK:

Python is one of the most popular programming language choices for implementing the back-end of web applications. Within the Python community, Flask is a very popular web framework. It is a micro-framework which implements a bare-minimum web server with simplicity and flexibility.

It only implements the core functionality giving developers the flexibility to add the feature as required during the implementation. It is a lightweight, WSGI application framework. This framework can either be used for pure backend as well as frontend if need be. The former provides the functionality of the interactive debugger, full request object, routing system for endpoints, HTTP utilities for handling entity tags, cache controls, dates, cookies etc. It also provides a threaded WSGI server for local development including the 9 test client for simulating the HTTP requests. Werkzeug and Jinja are the two core libraries. Since Flask is often termed as a prototyping framework, it does not include the abstraction layer for the database or any sorts of validation and security whatsoever. Therefore, Flask has given full flexibility to the implementor to add the requirements. There are extensions available for the Flask frameworks. Libraries but not limited to are, gunicorn for server, SQLAlchemy for database, Alembic for database migration management, celery & Redis for an asynchronous task runner, Flask-WTF form for form validation and Flask-limiter for rate-limiting the web requests. Flask is available for Python 3 and the newer version; it is also available in PyPy and easily installable with Python's official package manager pip.

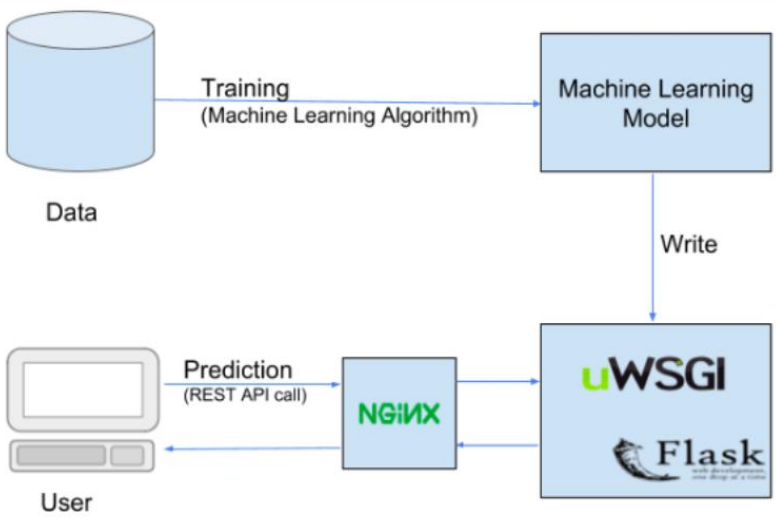



Fig 3.7

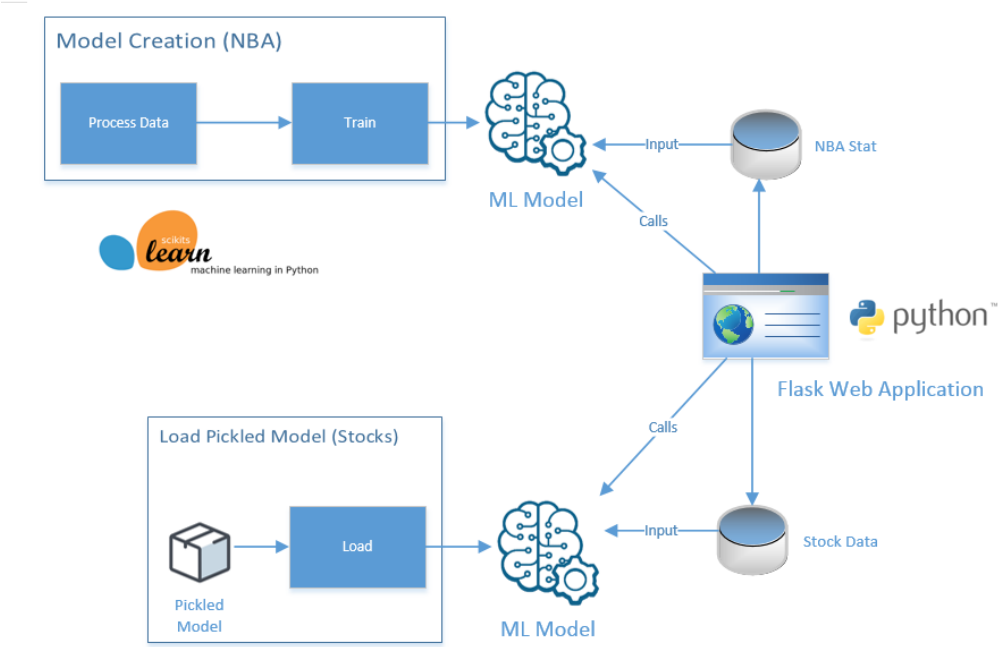



Fig 3.8

## 3.7 POSTGRES SQL:

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES. PostgreSQL is an open-source descendant of this original Berkeley code.

PostgreSQL adopts classical C/S structure, namely the daemon model that a client corresponds to a server. This daemon analyzes query requests coming from client, generates programming tree, retrieves data and lastly outputs formatting result to client.

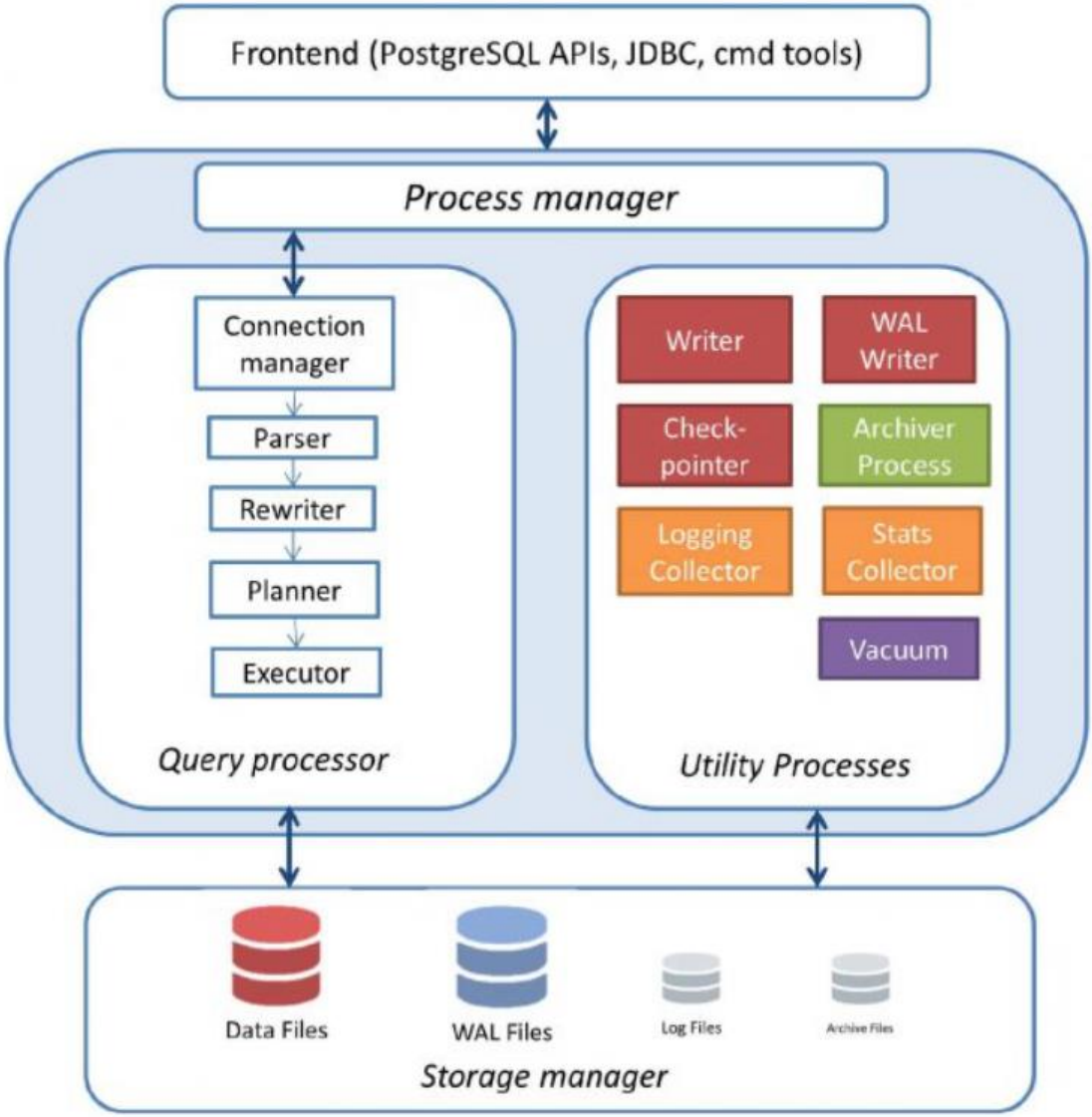
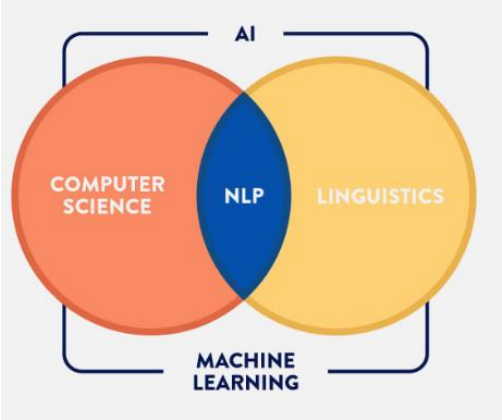



Fig 3.9

## 3.8 NATURAL LANGUAGE PROCESSING:

Natural Language Processing is defined as a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.




3.10

The goal of NLP is "to accomplish human-like language processing". Earlier, NLP was often termed as Natural Language Generation, it is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to:

- Text Parapharasing
- Translation of Text
- Provide answers to questionable content
- Automated inference to text

## 3.9 NATURAL LANGUAGE GENERATION:

Natural Language Generation is the subfield of artificial intelligence which involves automatically generating written texts in human languages, often from non-linguistic input data.

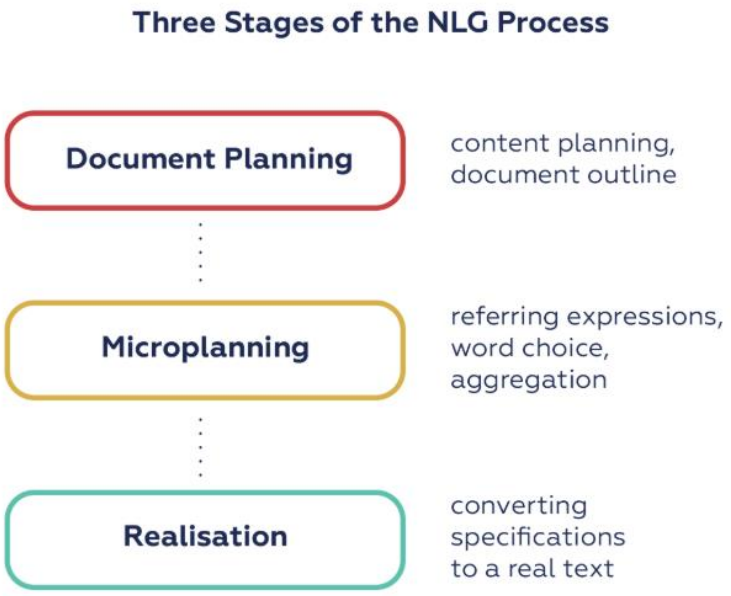



Fig 3.11

Nlg systems require types of knowledge to execute tasks. Nlg systems need domain knowledge (similar to that needed by expert systems), communication knowledge (similar to that needed by other Natural Language Processing systems), and also domain communication knowledge (DCK). DCK is knowledge about how information in a domain is usually communicated, including standard document structures, sublanguage grammars, and specialised lexicons.


Fig 3.12

At the same time, NLG has more theoretical applications that make it a valuable tool not only in Computer Science and Engineering, but also in Cognitive Science and Psycholinguistics. These include:


Fig 3.13

## 3.10 METRICS USED FOR EVALUATION:

### 3.10.1 Know Your Customer:

Know-Your-Customer (KYC) is a process which helps in understanding the nature of a customer's activities and to assess risks (if any) involved with the customer. Every on-boarding customer requires this process legally to establish customer identity. Currently, KYC is done individually by every business and the same data is provided by the users to multiple businesses and independently verified by each of them.


Fig 3.14

The process starts, when a financial institution is approached by a potential customer, who intends to work with it. The customer will hand in basic identity information and documents which are used by the financial institution to check for illicit activities. This results in an internal document which serves as certificate that the KYC process has been properly conducted and confirms whether the application has been approved or rejected. This process is necessary every time a new potential customer intends to work with a financial institution.

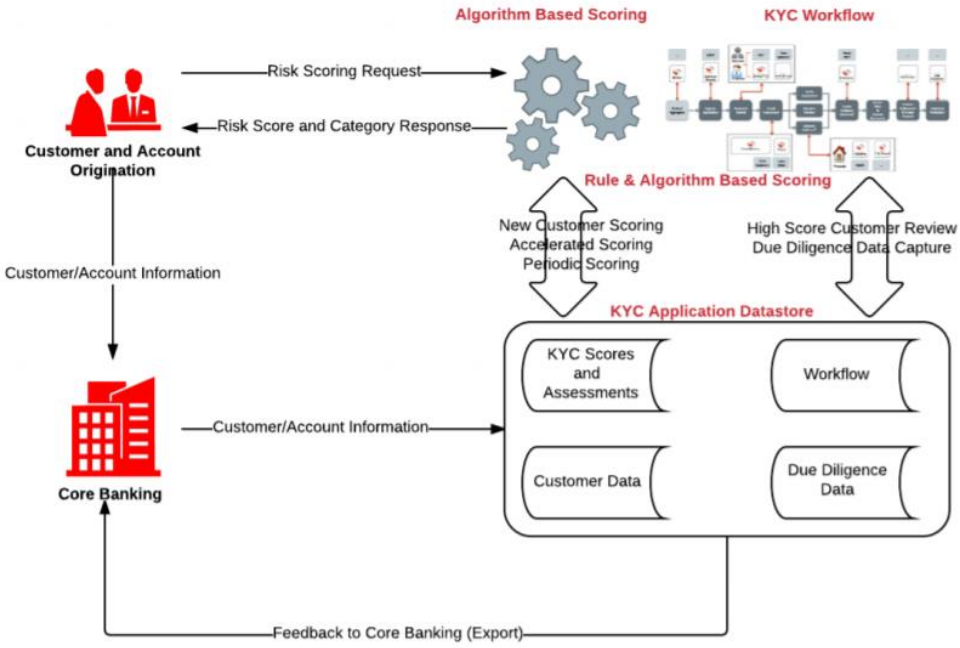



Fig 3.15

**3.10.2 EU Sanction List:**

Sanctions are an important tool of governance in the global financial industry intended to foreign relations, peacekeeping and conflict resolution. Most countries have used sanctions or had sanctions placed against either them or their citizen to fight economically, rather than physically.

EU sanctions are issued by the European Council. Every member of the council must agree on the sanction measures unanimously before legislation can be drafted that puts them into legal effect.

The EU Sanctions List is a consolidated list of countries, entities, and individuals, engaged in or suspected of money laundering or terrorism financing activities – and therefore subject to economic sanctions by the European Union. EU Sanctions are linked to United Nations Security Council Resolutions.

Sanctions imposed by the EU apply to financial institutions and individuals within the territory or jurisdiction of the European Union. Sanctions also apply to EU citizens operating outside EU territory. To ensure compliance, obligated financial institutions must integrate an EU sanctions search as part of their AML/CFT program when on boarding new customers. Failure to comply may trigger financial penalties and criminal charges against responsible individuals. The EU does not perform enforcement activities itself but delegates those actions to the relevant authorities in each member-state.

### 3.10.3 Politically Exposed Person:

A politically exposed person (PEP) is defined by the Financial Action Task Force (FATF) as an individual who is or has been entrusted with a prominent public function. Due to their position and influence, it is recognised that many PEPs are in positions that potentially can be abused for the purpose of committing money laundering (ML) offences and related predicate offences, including corruption and bribery, as well as conducting activity related to terrorist financing (TF). They have a higher risk of corruption due to their access to state accounts and funds.

Every EU-domiciled financial institution has to comply with the latest PEP regulations in place, and specifically in full compliance with the European legislation.

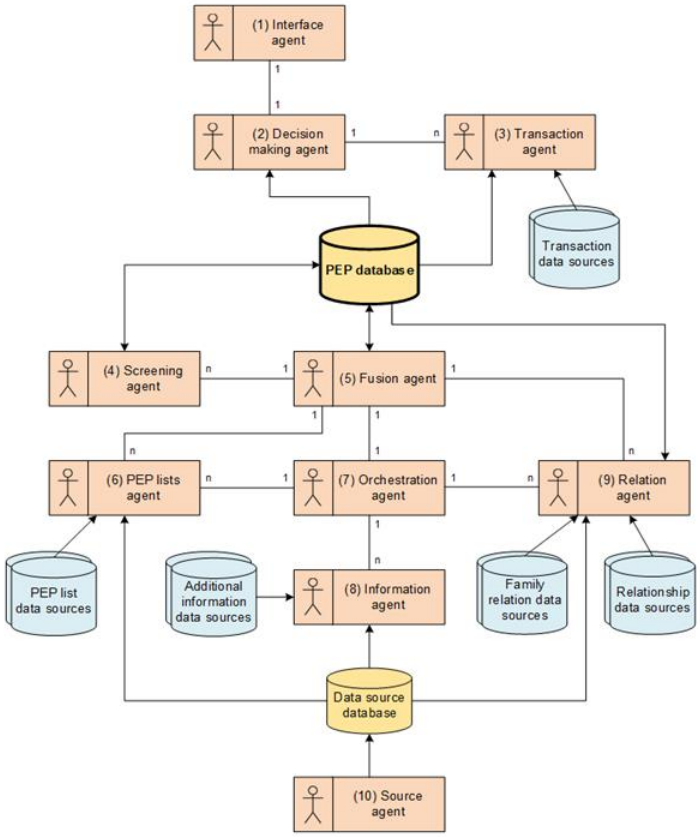



Fig 3.16

## 3.11. MACHINE LEARNING

Machine Learning is a type of Artificial intelligence that allows the system to learn without being explicitly programmed. The decisions in Machine Learning are driven by data rather than algorithms and also change its behavior, upon accommodating new information, that sets' it apart from the lot of technologies.

In the last decade, machine learning has introduced us to self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Today, Machine learning is so pervasive that you probably use it dozens of times a day without knowing it.

Machine learning and data mining process is similar. Both systems perform a search in any data to look for patterns. However, instead of human understanding of the extracted data - as in the case under the application of data mining - machine learning aims to use the data to detect patterns in the data, and adjust the plan of action accordingly.

### 3.11.1 Types Of Machine Learning Probelms:

The problem classes below are common for most of the problems when we are doing Machine Learning.

Classification: Data is labeled, for example spam/nonspam or fraud/non-fraud. The decision being modeled is to assign labels to new unlabeled pieces of data. This can be thought of as a discrimination problem, modeling the differences or similarities between groups.

Regression: Data is labeled with a real value (think floating point) rather than a label. Examples that are easy to understand are time series data like the price of a stock over time, the decision being modeled is what value to predict for new unpredicted data.

Clustering: Data is not labeled, but can be divided into groups based on similarity and other measures of natural structure in the data. An example from the above list would be organizing pictures by faces without names, where the human user has to assign names to groups, like iPhoto on the Mac. Rule

Extraction: Data is used as the basis for the extraction of propositional rules (antecedent/consequent aka if-then). Such rules may, but is typically not directed, meaning that the methods discover statistically supportable relationships between attributes in the data, not necessarily involving something that is being predicted.

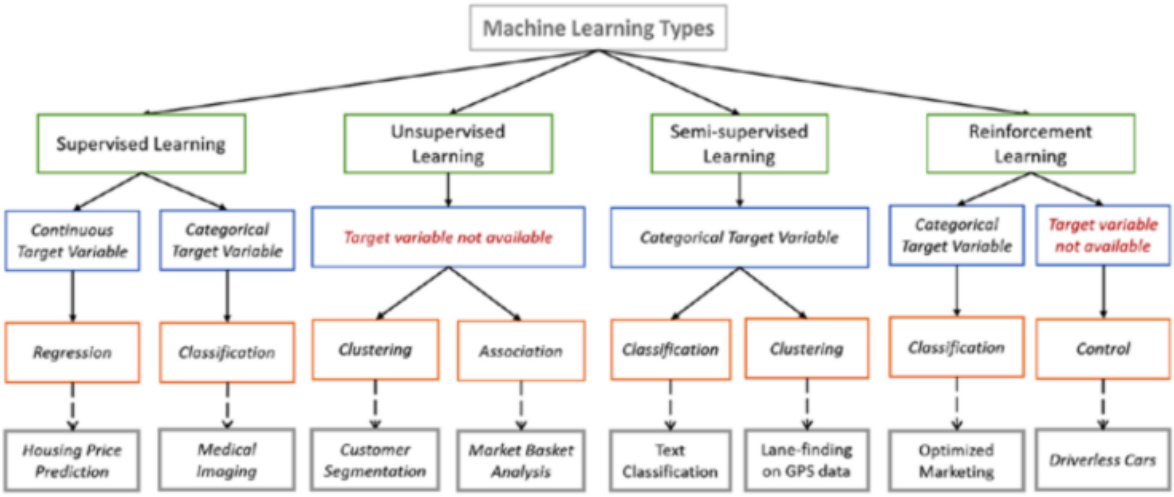

Machine Learning Types
Supervised Learning
Unsupervised Learning
Semi-supervised Learning
Reinforcement Learning
Continuous Target Variable
Categorical Target Variable
Target variable not available
Categorical Target Variable
Categorical Target Variable
Target variable not available
Regression
Classification
Clustering
Association
Classification
Clustering
Classification
Control
Housing Price Prediction
Medical Imaging
Customer Segmentation
Market Basket Analysis
Text Classification
Lane-finding on GPS data
Optimized Marketing
Driverless Cars

Fig 3.17

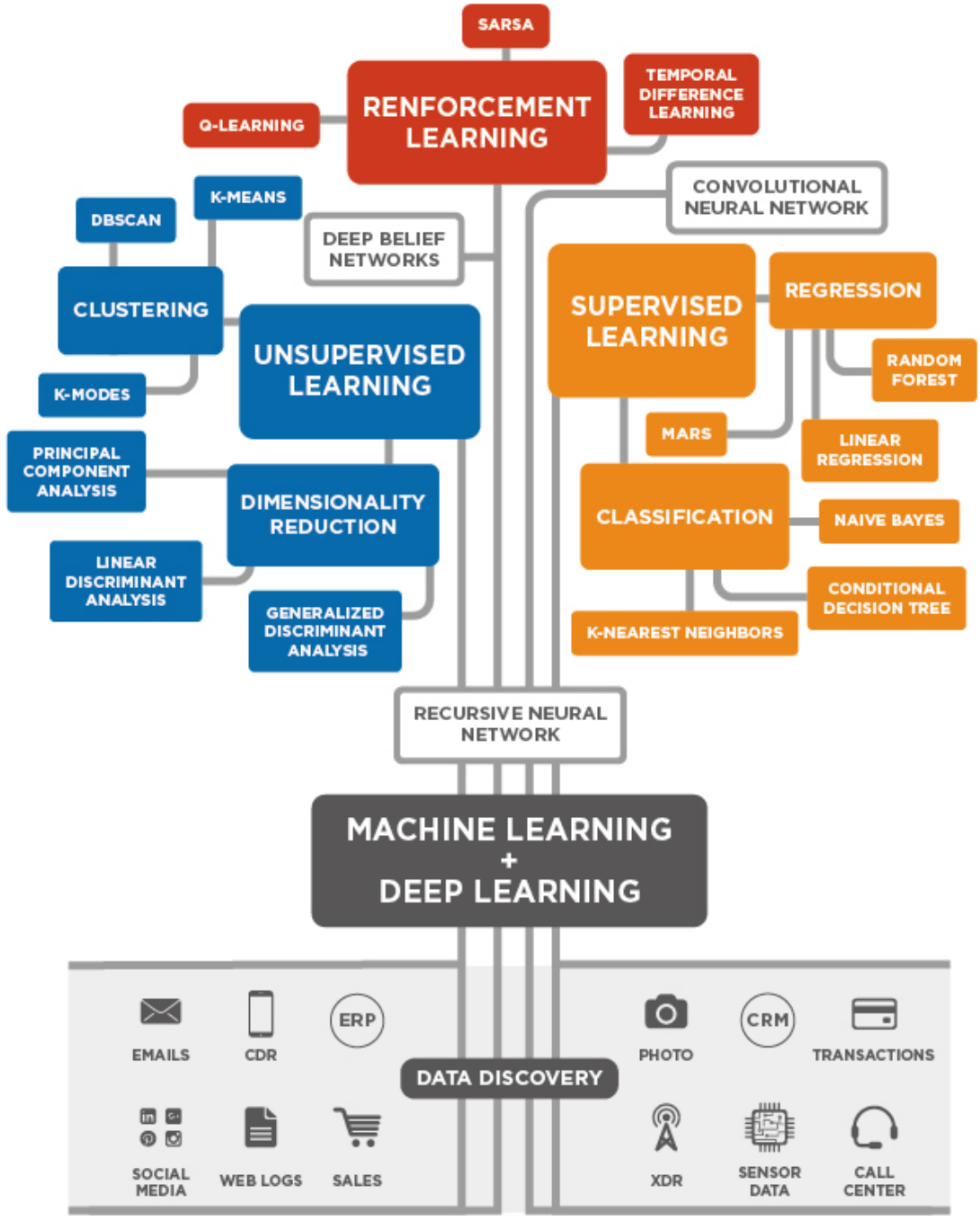



Fig 3.18

### 3.11.2 Applications of Machine learning:

There are numerous applications of machine learning. It's actually hard to realize how much machine learning has achieved in real world applications. Machine learning is

generally just a 64 way of fine tuning a system with tunable parameters. It is a way of making a system better with examples, usually in a supervised or unsupervised manner. Machine learning is normally applied in the offline training phase. Thus machine learning is used to improve the following applications.

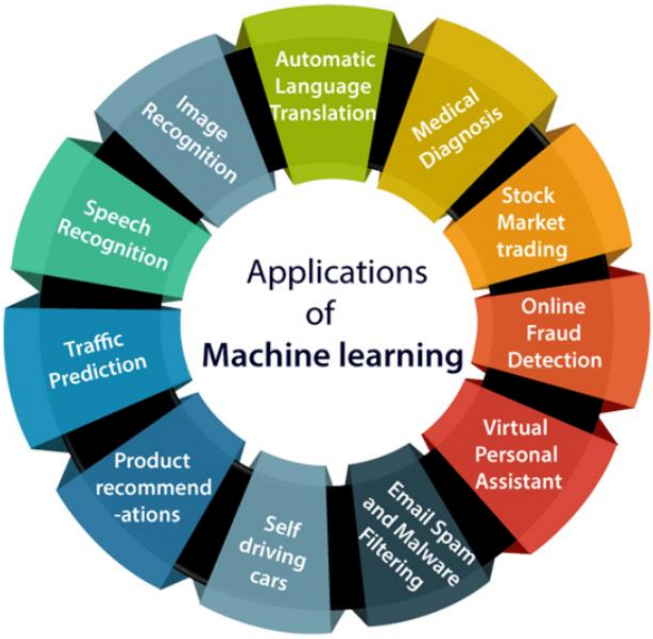



Fig 3.19

**MULTIPLE LINEAR REGRESSION:**

The multiple linear regression explains the relationship between **one continuous dependent variable** (*y*) and **two or more independent variables** (*x*1, *x*2, *x*3… etc)**.**

The following formula is a multiple linear regression model.

$$\mathbf{Y = B_0 + B_1X_1 + B_2X_2 + \ldots..BpXp}$$

Where:

**X, $X_1$, Xp** – the value of the independent variable,

**Y** – the value of the dependent variable.

$\mathbf{B_0}$ – is a constant (shows the value of Y when the value of X=0)

$\mathbf{B_1}$, $\mathbf{B_2}$, **Bp** – the regression coefficient (shows how much Y changes for each unit change in X)

## RIDGE REGRESSION:

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. Multicollinearity, or collinearity, is the existence of near-linear relationships among the independent variables. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.

Regression coefficients are estimated using the formula

$$\hat{\underline{B}} = (X'X)^{-1}X'\underline{Y}$$

X'X = R, where R is the correlation matrix of independent variables.

## LASSO:

Lasso solutions are quadratic programming problems, which are best solved with software. The goal of the algorithm is to minimize:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

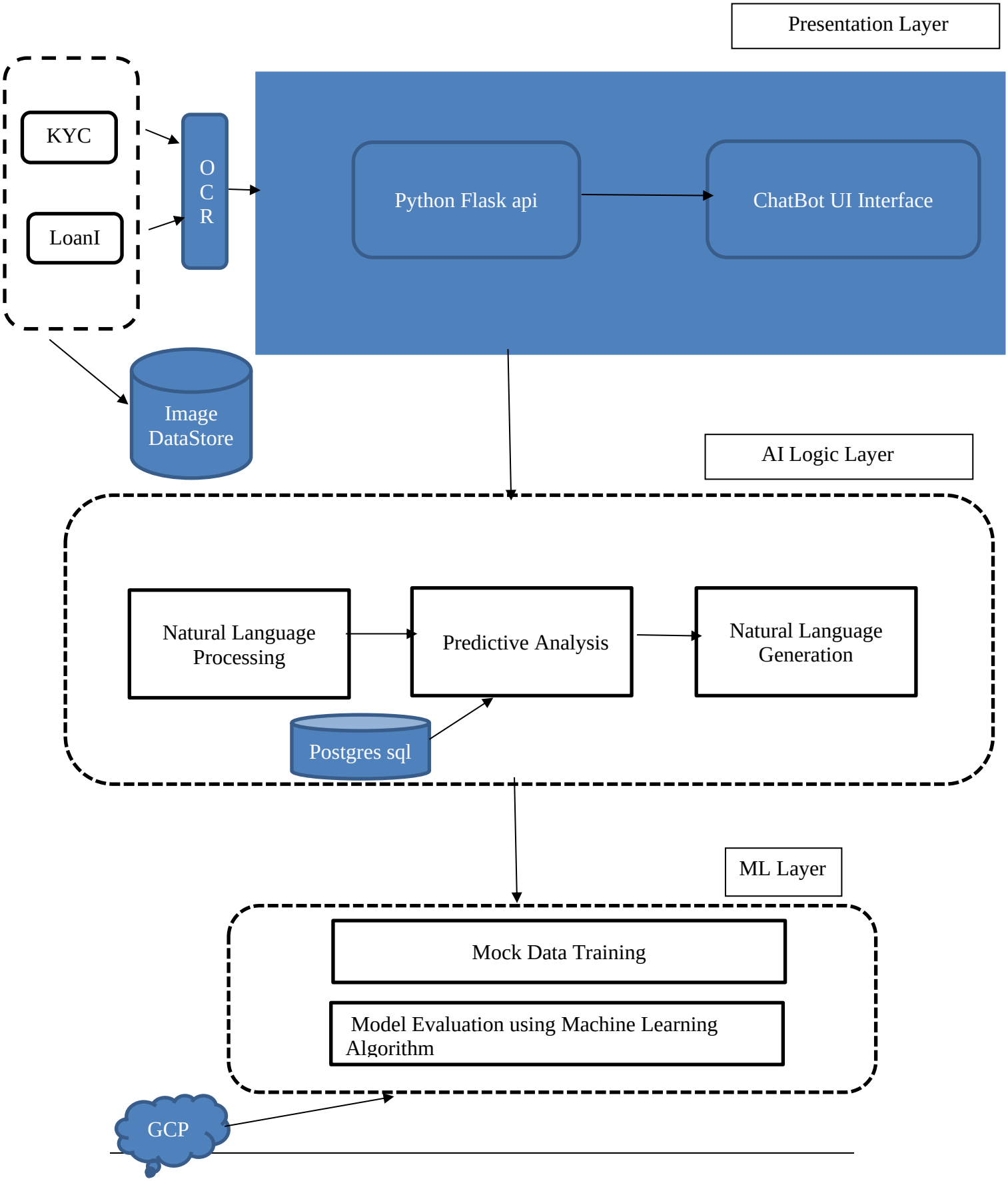
Which is the same as minimizing the sum of squares with constraint $\Sigma |B_j \leq s$. Some of

the βs are shrunk to exactly zero, resulting in a regression model that's easier to interpret.

# Chapter 4. System Design and Specifications

This chapter analyses the various components of the Checker, starting from the Presentation layer, followed by the Machine Learning layer and the AI layer. A deep dissection of each layer, based on the various components part of it is done.

**4.1 Architectural Diagram:**



**4.1.1 Presentation Layer:**

**Optical Character Recognition:**

- The core efficiency of the model depends up on the accuracy of the OCR process, thus making it the most pivotal component of the application.
- The user is prompted to upload the front page of the passport.

**Know Your Customer Data:**

The KYC data is determined by the Optical Character recognition of the Passport Page uploaded by the user. It is done by scanning the mrz code of the passport.

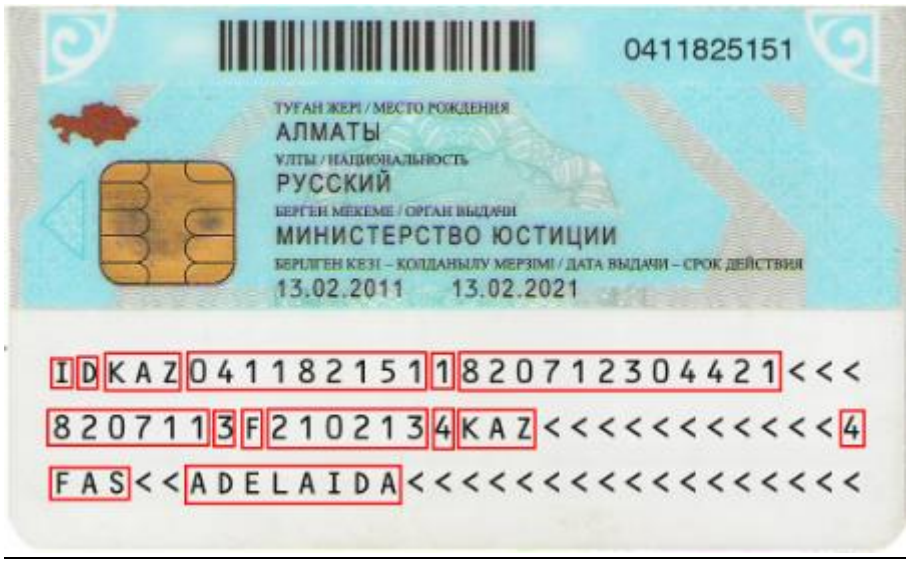



Fig 4.1

Based on the mrz code, various details of the user are scanned in its absolute raw state. Those improper data are further transformed into a workable format.

**LoanIQ:**

- Next up the loanIQ is fed in by the user.
- This is done to channel the user based on his location and the loan preference.
- In the case of this application, the loan IQ data is considered to be the Address and the Place of birth of the user.

**ChatBOT Interface:**

In-order to provide the user with an interactive experience, a chat bot is built. The chat bot is built based on the training data that is provided for the banking purpose that is to be served.

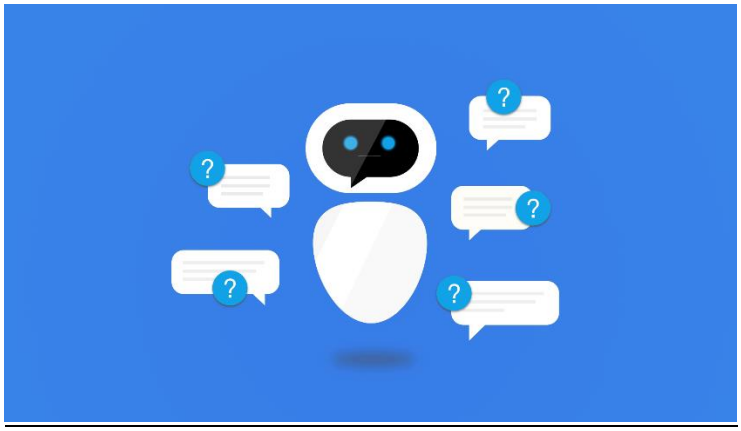


Fig 4.2

Based on the loan preference of the user, the chatbot is built in such a way that the user is channeled into multiple channels. Once the backend process is done, the user is provided with a NLG narrative through the chatbot.

**Flask API:**

The chatbot is developed using the Python Flask API. Once the user starts interacting with the Chatbot, the user based on his loan preference is streamlined and re-directed through to the OCR application where the screening happens.

**AI Logic Layer:**

In this layer, the following MRZ code is analyzed using NLP and the results are provided using the Natural Language Generation.

```
P<GBRNASH<<STEVE<<<<<<<<<<<<<<<<<<<<<<<<<<<<
G100519738GBR9702112M1807136<<<<<<<<<<<<<<04
```

Fig 4.3

Once the MRZ code is scanned and converted to workable format, stored in the google cloud platform. The Predictive Analysis is executed and stored in the cloud.

**Machine Learning Layer:**

In this layer, various machine learning algorithms are implemented to the mock data generated and using Predictive Analysis the efficiency of the model is Improved. The mock data is generated based on 40 actual data from the sanction list and 60 random data. It is stored in the cloud for further use.

Also, further the evaluation of the efficiency of the machine is also validated in this layer.The evaluation is based on the data retrieved as the threshold frequency, which is stored in the Postgres SQL.

Further the model is validated using various Regression Model such as the:

- Multiple Linear Regression
- Ridge Regression’
- Lasso Regression
- Elasticnet Regression

# Chapter 5. Implementation

## 5.1 WorkFlow:

### 5.1.1 Sanction List Pre-processing:

The sanction list is loaded into the system by downloading from the EU financial Sanction list website. Using a HTTP request the list is downloaded in the csv format. Further the data is converted into a dataframe in python using the following libraries:
import requests: The Requests library allows the user to send HTTP/1.1 requests to the web easily. The query strings can be either added to the URLs, or as GET OR POST request.
import shutil : It is used to import either single or collection of files.
Sample Sanction List:

```
In [12]: sanction.columns = ['First Name','Last Name','WholeName','Date of Birth','Address','Country','City','Passport Number','Country C
```

```
In [13]: sanction
```

Out[13]:

| | First Name | Last Name | WholeName | Date of Birth | Address | Country | City | Passport Number | Country Code | Gender | Issue Date | Expiry Date | Place of Issue | Place of Birth |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Saddam | Hussein Al-Tikriti | Saddam Hussein Al-Tikriti | | | | | | | M | | | | |
| 1 | | | Abu Ali | | | | | | | | | | | |
| 2 | | | Abou Ali | | | | | | | | | | | |
| 3 | | | | 1937-04-28 | | | | | | | | | | al-Awja near Tikrit |
| 4 | | | | | | | | | | | | | | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 12935 | | | | 1976-10-01 | | | | | | | | | | Tallâ€™Afar |
| 12936 | | | | 1976-10-01 | | | | | | | | | | Mosul |
| 12937 | | | | 1976-10-05 | | | | | | | | | | Mosul |
| 12938 | | | | 1976-10-05 | | | | | | | | | | Tallâ€™Afar |
| 12939 | | | | | | | | | | | | | | |

12940 rows × 14 columns

Fig 5.1

### 5.1.2 Store Sanction List in GCP:

After downloading the list, it is stored in the Google Cloud Virtual Storage. It is done by providing the authorizations of the credentials to the Google bucket. Further numerous libraries of python are used for such process.

import oauth2.service: It is for as a client library as an authorization service.

```
credentials_dict = {
  "type": "service_account",
  "project_id": "aki-python",

  "private_key_id": "73202949799976fc76d72d6d38b3935129315a17",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDADlmjF3ls1Bv/\nbneunQNRnhpN6+X
  "client_email": "akilesh7910@aki-python.iam.gserviceaccount.com",
  "client_id": "114127699863693119653",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/akilesh7910%40aki-python.iam.gserviceaccount.com"
}
```

Fig 5.2

Import gcloud: It is used as a client, to access the Google Cloud Platforms.

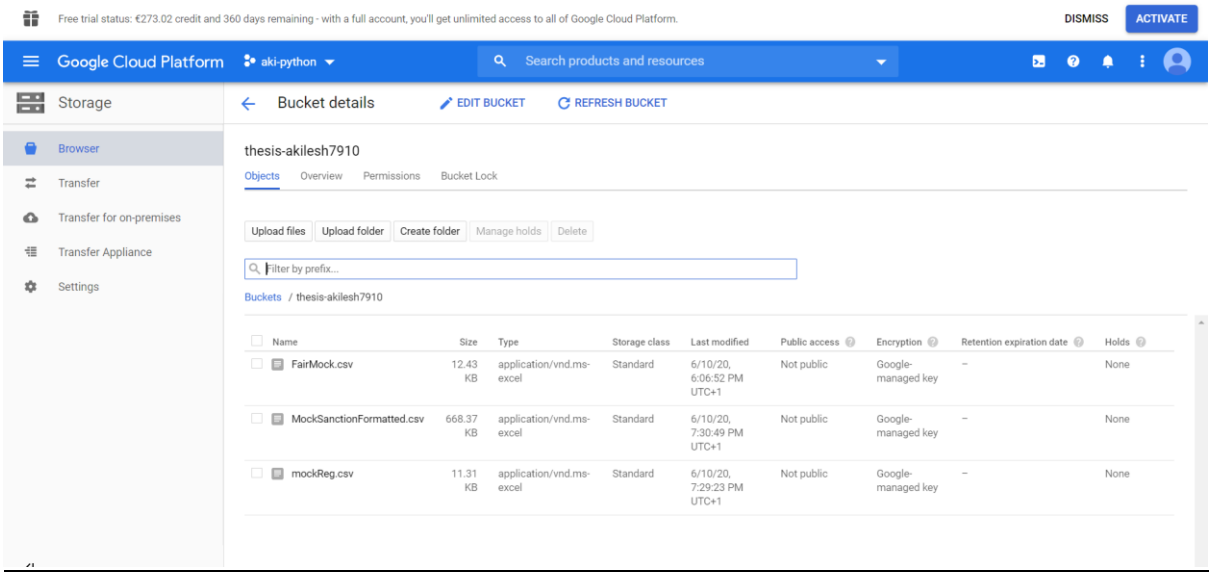



Fig 5.3

## 5.2 USER INTERFACE:

### 5.2.1 Chatbot:

The Chatbot is designed in such a way that the user is allowed to have an interactive experience with the application. The chatbot training is done both manually and through using a pre-defined corpus.
Using the corpus, the chatbot is allowed to have common gestures of greeting and other social etiquettes. The libraries used to build a chatbot are:
import chatterbot: engine built in python that allows it to generate responses based on collection of pre-defined conversations. It is based on machine language algorithm.
import ChatterBotCorpusTrainers: This is used to access the pre-defined chatter bot corpus of data, that is used to train the speech pattern of the Bot.

### 5.2.2 OCR-Passport Dissection:

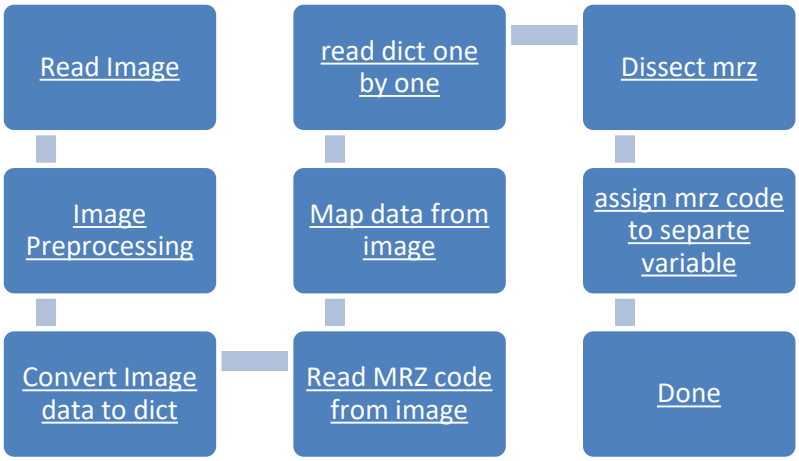
**Libraries Accessed:**

Pytesseract is an image wrapper from Google's Tesseract OCR Engine. It is useful in inovaction of script to tesseract, that makes itself read images of all types such as the Pillow and the Leptonica imaging libraries, including jpeg, png, gif.etc. If it is used as a standalone script the then the pytesseract recognizes the text and print it instead of writing it to a file.
PassportEye: This package acts as a tool that recognizes machine readable zones (MRZ) from scanned id documents. Once the documents are be located rather arbitrarily on the page - the code tries to find anything resembling a MRZ and parse it from there.
CV2: Pre-built OpenCV packages for Python
Numpy: It is a package used for undergoing fundamental scientific operations using Python

**Operation:**

The Pressure point of the entire project depends on getting accurate KYC data from the customer. In this case the KYC document used for ID proof is the Passport. By scanning the Passport, basic information about the customer applying for the loan is fetched by the bank. This process involves the following steps:

- The scanned image of the passport is uploaded by the user to an UI developed using python. Then, the image is measured for quality and once the quality is checked the image is pre-processed.
- In this case, the grayscale and the deskew checks of the image are done once before scanning.
- Then once the image is pre-processed, using the OCR (Optical Character Recognition) library Tesseract, the image is analyzed and the data is retrieved from the image. The mrz code of the passport is used for this purpose and by using the PassportEye library the name, DOB, passport expiry date, passport number and the nationality is mapped.




**Use Case:**

Fig 5.4

From the following Mock Passport, once uploaded the MRZ data is obtained. Each data of the mrz data is stored in a Dataframe and is checked for similarity of it's content with the Sanction List.

## 5.3 Algorithm Formulation:

### 5.3.1 Similarity Algorithm:

The Algorithm used to check for similarity between the OCR data and the sanction list is the Levenshtein's Algorithm. It is based on converting the data into a matrix and then weighs and distance and the cost between each element of the matrix. It generates the similarity between two words based on its distance from each other.

### Calculation of Distance:

```
            distance[row][col] = min(distance[row-1][col] + 1,      # Cost of deletions
                                     distance[row][col-1] + 1,      # Cost of insertions
                                     distance[row-1][col-1] + cost)     # Cost of substitutions
    if ratio_calc == True:
        # Computation of the Levenshtein Distance Ratio
        Ratio = ((len(s)+len(t)) - distance[row][col]) / (len(s)+len(t))
        return Ratio
```

Fig 5.5

### Sample:

```
Str1 = "UNITED STATES"
Str2 = "UNKNOWN";
Distance = levenshtein_ratio_and_distance(Str1.lower(),Str2.lower())
print(Distance)
Ratio = levenshtein_ratio_and_distance(Str1.lower(),Str2.lower(),ratio_calc = True)
print(Ratio)
```

```
The strings are 11 edits away
0.2
```

Fig 5.6

### 5.3.2 Algorithm to generate Risk Score:

Based on the MRZ data, along with the input address and Place of Birth data provided by the user the, the risk score is generated based on the similarity of the User data with the sanction list.

| Data1 | Threshold Frequency | Risk Score Constitutent(Weightage)(tent.) |
|---|---|---|
| WholeName: | f1 | 0.166 |
| FirstName | f2 | 0.056 |
| LastName | f3 | 0.056 |
| Data2 | | |
| Address: | f4 | 0.166 |
| City | f5 | 0.056 |
| Country | f6 | 0.056 |
| Place of birth | f7 | 0.056 |
| Data3 | | |
| passport number: | f8 | 0.166 |
| issue date | f9 | 0.056 |
| expiry date | f10 | 0.056 |
| Place of issue | f11 | 0.056 |
| Country Description | f12 | 0.056 |

Fig 5.7

Based on the type of input data, the weightage for each is mentioned. The risk score(n) is incremented by its weightage score every time the input data surpasses its Threshold Frequency. The Threshold Frequency is discussed among further chapters.

(if Data1, Data2, Data3>Threshold Frequency(fi)):n++

### 5.4 Mock Data Generation:

- In-order to generate the threshold frequency score for each attribute, mock data worth of 100 entries was generated.
- It was generated in such a way that 40% of the dataset was randomly taken from the sanction while rest 60% was made of random data.
- Using that the threshold frequency f1, f2….f12 was generated.

  Sample:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 30 | Abu | DORDA | Abu Zayd U | ######## | | LIBYA | | FK117RK0 | LBY | | ######## | ######## | LBY | | | |
| 31 | Abdelkebir | Fakhiri | Abdelkebir | ######## | | | | B/014965 | | | | | LBY | | | |
| 32 | Mohamed | Mubarak | Mohamed | ######## | | | | | | M | | | | | | |
| 33 | Mohamed | Hamcho | Mohamed | ######## | | | | 2954347 | | | | | SYR | | | |
| 34 | Qasim | Soleimani | Qasim Sole | ######## | | | | "008827 | | | | | SYR | Qom | | |
| 35 | Mohamma | Mufleh | Mohammad Mufleh | | | | | | | M | | | | | | |
| 36 | | | Heris Syah | ######## | Jalan M. Sa | INDONESI | South Jakarta | | TAQA | | | | | East Lombok West Nusa Tenggara | | |
| 37 | | | Issam Anbouba | | | | | | | M | | | | Homs | | |
| 38 | | | Abu Mansu | ######## | | SOMALIA | | 40306256 | SOM | | | | SOM | ALABAMA | | |
| 39 | Seyed | FIROUZAB | Seyed Has | ######## | | | | | | M | | | | Mashhad | | |
| 40 | Muhamma | Iqbal | Muhamma | ######## | 4 Masjid al | PAKISTAN | Lahore | DG514948 | TAQA | | | | | | | |
| | Porfirio | Lanier | Porfirio La | 11-19-196 | 1 3RD PL U | Argentina | Chaco | 1327M3YY | ARG | F | ######## | ######## | ARG | Chaco | | |
| | Reece | Laurent | Reece Lau | ######## | 1 AVIGNOI | Argentina | Buenos Air | F8624KW3 | ARG | F | ######## | 01-22-201 | ARG | Buenos Aires | | |
| | Rich | Lemar | Rich Lema | ######## | 1 BLACKSV | Argentina | Santa Fe | G730Z-H5 | ARG | F | 07-16-200 | ######## | ARG | Santa Fe | | |
| | Ruth | Lex | Ruth Lex | 04-29-196 | 1 BLUE HC | Argentina | Misiones | F424R4W9 | ARG | M | 01-30-199 | 04-17-201 | ARG | Misiones | | |
| | Sherwin | Manning | Sherwin M | ######## | 1 BRISA DE | Argentina | Neuquen | C990TNM | ARG | M | ######## | 08-18-201 | ARG | Neuquen | | |

Fig 5.8

The mock dataset is stored in the Google Cloud Platform for future access purpose.

### 5.4.1 Mock Data Training:

Once the dataset is generated, each entry of the dataset is checked for similarity using the Levenshstein algorithm for all attributes. Then average score of all hundred entries for each entire is calculated and it is taken as the Threshold Frequency for each attribute.

```
[0.7291496349607942,
 0.802059149340789,
 0.7698455245504683,
 0.5949602953219469,
 0.5851222936780848,
 0.7529463458282198,
 0.4760411678733651,
 0.8808080808080798,
 0.68013468013468,
 0.666666666666667,
 1.0,
 0.8757575757575751]
```

Fig 5.9

All the values are then stored in a Postgressql. It is done by connecting the Python Editor(Jupyter Notebook) and storing the data as a table in the database. The authentication is required to access the data. The libraries imported to access the

database is sqlalchemy, from which the data can be queried and in regards to the authorization of the database, psycopy2 library is imported.

**Credentials for the Database:**

```
try:
    connection = psycopg2.connect(user="Akilesh",
                                  password="7910",
                                  host="127.0.0.1",
                                  port="5432",
                                  database="reg_scoreDB")
    cursor = connection.cursor()
    postgreSQL_select_Query = "select * from reg_table"

    cursor.execute(postgreSQL_select_Query)
    print("Selecting rows from mobile table using cursor.fetchall")
    mobile_records = cursor.fetchall()
```

Fig 5.10

**Library Accessed:**

Psycopg2: This library acts as a postgresql adapter.

Sqlalchemy: This library is used to access sql queries to fetch data from the database.

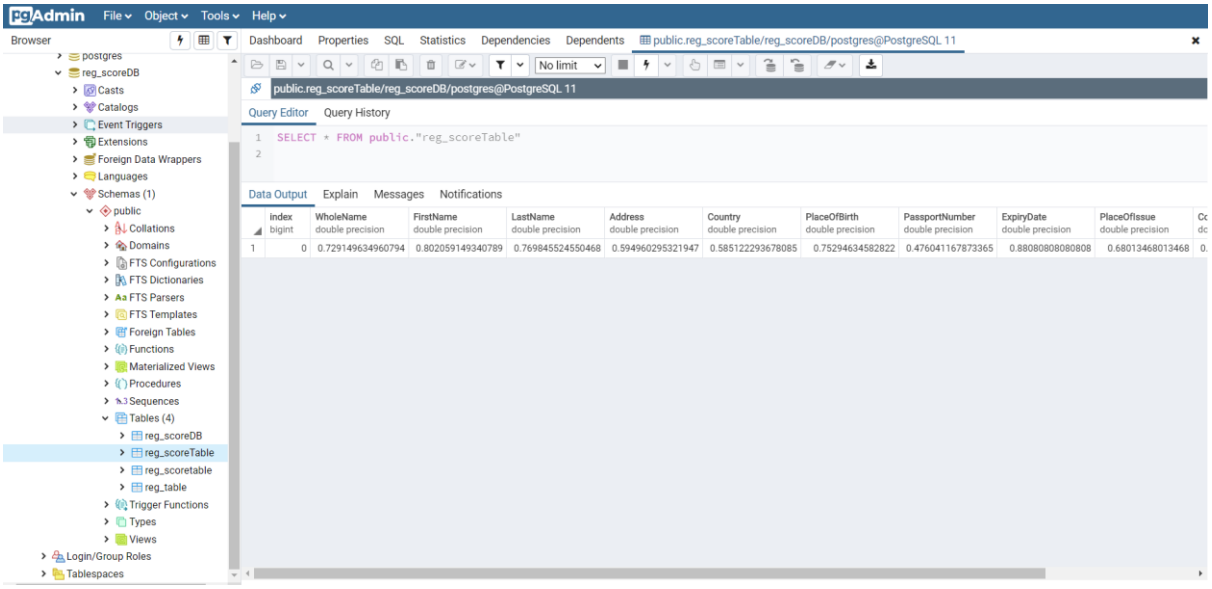



Fig 5.11

The above image is the visual confirmation of the database n=being stored in the postgresql database. The visualization is done using PGadmin4.

Out[207]:

| | FirstName | LastName | WholeName | Address | Country | PlaceOFBirth | PNumber | ExpiryDate | PlaceOFI | CountryCode | Gender | Dob | risk_score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.380952 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.500 |
| 1 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.526316 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.834 |
| 2 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.380952 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.500 |
| 3 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.834 |
| 4 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.834 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 94 | 0.800000 | 0.545455 | 0.608696 | 0.285714 | 0.222222 | 1.000000 | 0.244898 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.166 |
| 95 | 0.833333 | 0.571429 | 0.571429 | 0.327869 | 0.260870 | 1.000000 | 0.160000 | 0.8 | 0.666667 | 0.666667 | 1.0 | 0.9 | 0.166 |
| 96 | 0.571429 | 0.666667 | 0.538462 | 0.305882 | 0.250000 | 0.500000 | 0.300000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.9 | 0.000 |
| 97 | 0.588235 | 0.600000 | 0.551724 | 0.404762 | 0.307692 | 0.733333 | 0.400000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.000 |
| 98 | 0.625000 | 0.666667 | 0.500000 | 0.367816 | 0.307692 | 0.755556 | 0.400000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.000 |

The similarity score for each mock data is applied.

## IX User Interface:Use Case

## Chatbot UI:

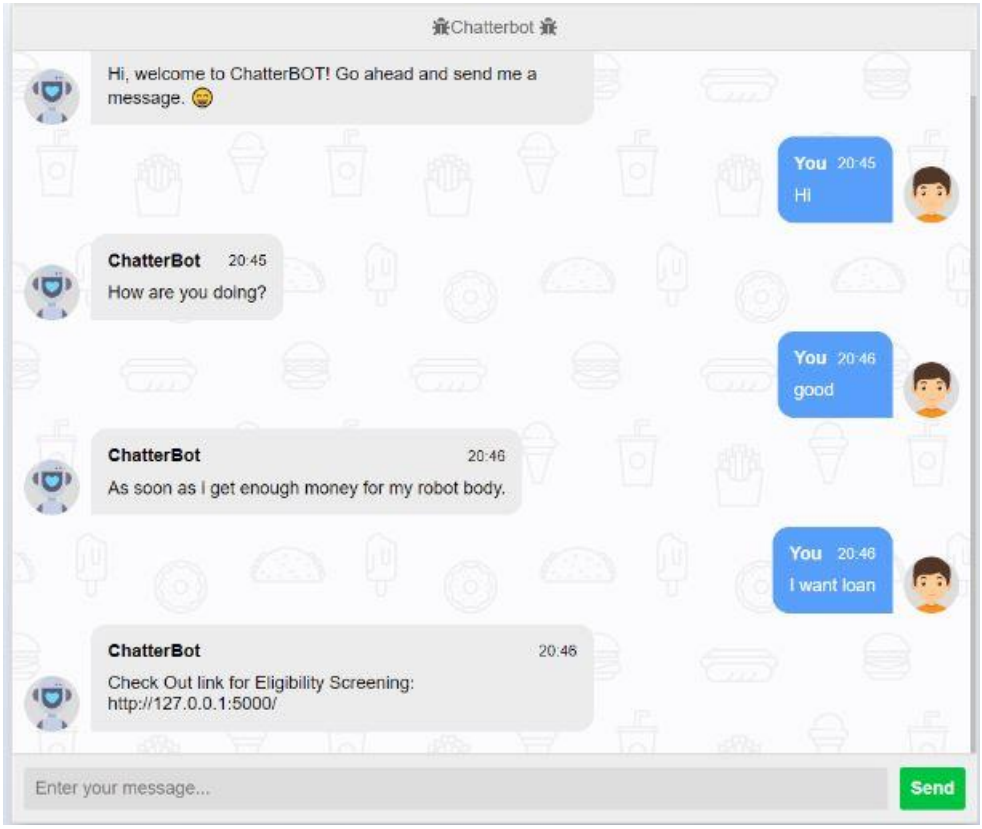



Fig 5.12

## OCR UI:

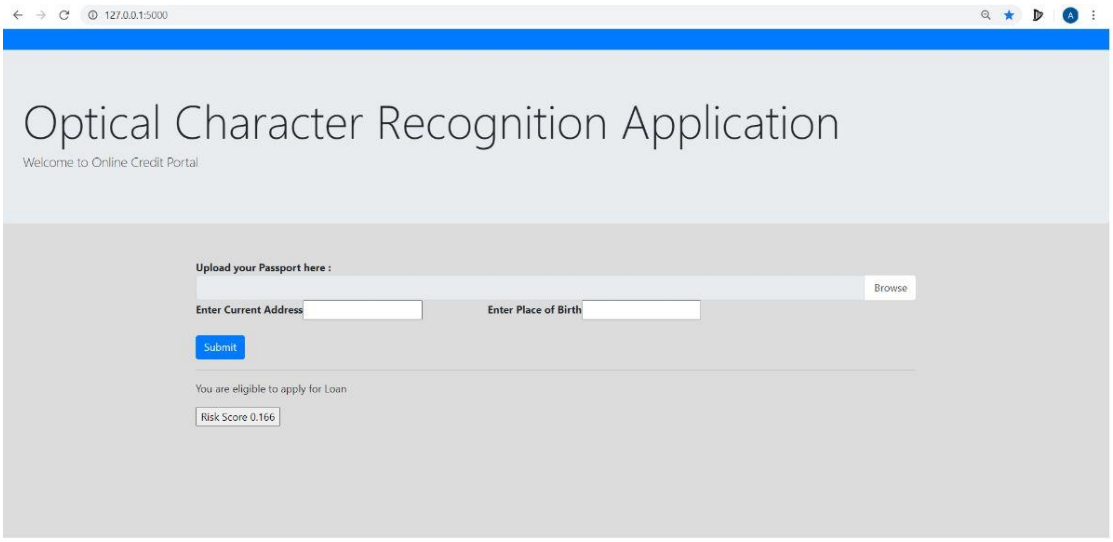



Fig 5.13

## X NLG Narrative:

Based on the risk score, the customer is provided with one of four responses.

```
NLG Narrative:

   Risk Score Category                        Narrative
  |
1  0-0.4- Less Liable

   Sub Category

   0.2-0.4: Upper Less Liable

2  0.4-0.7- Cautiouslsy Liable

   Sub Category

   0.4-0.6 Lower Caution
   0.6-0.7 Higher Caution

3  0.7>- High Liability

BOT Narrative:

Case1: 0-0.4 : "You are eligible to apply for Loan"
Case2: 0.4-0.6: "You can only apply for PPP loan at the moment"
Case3: 0.6-0.7 ": "You need to come in contact with representative for further vetting"
Case4: 0.7:"You are higly liable and ineligible to apply for loan"
```

Fig 5.14

# Chapter 6. Testing and Evaluation

In-order to check the efficiency of the model multiple regression machine learning algorithms are used. They are:

- Multiple Linear Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression.

**Scope of Data:**

The dataset contains the passport details of about 100 passengers. It has 12 independent variables and 1 dependent attribute. Attributes namely "FirstName", "LastName", "WholeName", "Address", "Country", "PlaceOFBirth", "PNumber", "ExpiryDate", "PlaceOFI", "CountryCode", "Gender" and "Dob" are taken as explanatory variables and the “Risk score” is considered to be the response variable. Since all has continuous values regression algorithms has been used and detailed.

## 6.1 Multiple Linear Regression:

Multiple linear regression model examines the relationship between a dependent variable and multiple independent variables. The risk score will be predicted that is implemented in python.

### 6.1.1. Defining the variables:

```
result = pd.concat([dataset, dataset1], axis=1, join='inner')
```

```
result
```

| | FirstName | LastName | WholeName | Address | Country | PlaceOFBirth | PNumber | ExpiryDate | PlaceOFI | CountryCode | Gender | Dob | risk_score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.380952 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.500 |
| 1 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.526316 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.500 |
| 2 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.380952 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.500 |
| 3 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.890 |
| 4 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 | 1.0 | 1.0 | 0.890 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 94 | 0.800000 | 0.545455 | 0.608696 | 0.285714 | 0.222222 | 1.000000 | 0.244898 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.166 |
| 95 | 0.833333 | 0.571429 | 0.571429 | 0.327869 | 0.260870 | 1.000000 | 0.160000 | 0.8 | 0.666667 | 0.666667 | 1.0 | 0.9 | 0.166 |
| 96 | 0.571429 | 0.666667 | 0.538462 | 0.305882 | 0.250000 | 0.500000 | 0.300000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.9 | 0.000 |
| 97 | 0.588235 | 0.600000 | 0.551724 | 0.404762 | 0.307692 | 0.733333 | 0.400000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.000 |
| 98 | 0.625000 | 0.666667 | 0.500000 | 0.367816 | 0.307692 | 0.755556 | 0.400000 | 0.8 | 0.333333 | 0.333333 | 1.0 | 0.8 | 0.000 |

99 rows × 13 columns

```
X = result[["FirstName","LastName","WholeName","Address","Country","PlaceOFBirth","PNumber","ExpiryDate","PlaceOFI","CountryC
Y = result[['risk_score']]
```

Fig 6.1

- As it has been detailed before, ‘dataset1’ dataframe contains only the risk score values of all hundred passengers and the rest (all predictor attributes) is stored in the ‘dataset’ dataframe.
- Using inner join two dataframes has been combined and is stored in a new dataframe called ‘result’.
- The ‘result’ dataframe has a combined set of columns from ‘dataset’ and ‘dataset1’. The resultant is displayed in the above figure.
- Now before fitting the regression models we will define X and Y variables i.e. all independent attributes from the dataframe in X variable and the target variable is defined as Y.

### 6.1.2 Loading the required libraries*:*

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

Fig 6.2

- Scikit-learn is a library that features multiple machine learning algorithms. Linear model has been imported to apply multi-linear regression model.
- Splitting dataset means separating them into train and test data ratio. We divide them into 80-20 ratio 80 percent is for training & validation and 20 percent dataset used for testing purpose.
- Python library ‘ScikitLearn’ is divides the data into training and testing set using “train_test_split” function as shown above.

### 6.1.3 Fitting the Model:

```
regressor = linear_model.LinearRegression()
regressor.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Fig 6.3

- Now the model is fitted with the model name ‘Linear regression’ as regressor a variable. The regressor.fit() function fits a multiple linear model and the same has made accurate predictions as follows.

### 6.1.4 **Predicted values:**

```
Y_pred = regressor.predict(X_test)
print(Y_pred[0:5])
```

```
[[ 0.87819837]
 [-0.09862736]
 [ 0.5294008 ]
 [ 0.34998739]
 [ 0.12396973]]
```

Fig 6.4

- The print function of Y_pred displays a sample of first 5 predictions of risk score of the test data.
- As explained, regressor.predict() predicts the risk score using the linear model we fitted.

```
res = regressor.score(X_train, Y_train)
print('Score:',res)
print('Intercept:', regressor.intercept_)
print('\nSlope:', regressor.coef_)
```

```
Score: 0.985620214336308
Intercept: [-0.71725561]

Slope: [[ 0.74698278  0.10262926 -0.1226391   0.068348    0.14964358  0.08954549
   0.56344223  0.07539148  0.07673229 -0.06987063  0.         -0.08475141]]
```

Fig 6.5

- With the help of using the built-in functions the score, the coefficients and the intercepts have been estimated for the training data as shown in the above figure.
- The score defines the $R^2$ value that is 98 percentage of explained variance of the predictions i.e. (subtracting predicted values from the actual values).
- Coefficients and intercepts for the predictors have also been checked and displayed.

### 6.1.4 Actual values Vs Predicted values:

```
res = pd.DataFrame({'actual':Y_test[:,0], 'predicted':Y_pred[:,0]})
res.head(100)
```

| | actual | predicted |
|---|---|---|
| 0 | 0.890 | 0.878198 |
| 1 | 0.000 | -0.098627 |
| 2 | 0.500 | 0.529401 |
| 3 | 0.166 | 0.349987 |
| 4 | 0.166 | 0.123970 |
| 5 | 0.166 | 0.223573 |
| 6 | 0.500 | 0.486239 |
| 7 | 0.000 | 0.024161 |
| 8 | 0.000 | -0.001607 |
| 9 | 0.166 | 0.136773 |
| 10 | 0.166 | 0.196965 |
| 11 | 0.000 | 0.127272 |
| 12 | 0.222 | 0.186421 |
| 13 | 0.890 | 0.878198 |
| 14 | 0.890 | 0.878198 |
| 15 | 0.500 | 0.502570 |
| 16 | 0.890 | 0.878198 |
| 17 | 0.890 | 0.878198 |
| 18 | 0.500 | 0.510736 |
| 19 | 0.222 | 0.497617 |

Fig 6.6

- Actual values of the test data (Y_test) is compared with the predicted values and the model has predicted more precisely.

### 6.1.4 Plotting univariate distribution (Risk score):

```
import matplotlib.pyplot as plt
import seaborn as seabornInstance
plt.figure(figsize=(6,5))
plt.tight_layout()
#seabornInstance.distplot(df['price'])
seabornInstance.distplot(result['risk_score'])
```

Fig 6.7

- Seaborn library is imported for visualising the distribution of the dependent variable – risk score.
- We have used distplot() function to present the histogram as follows:

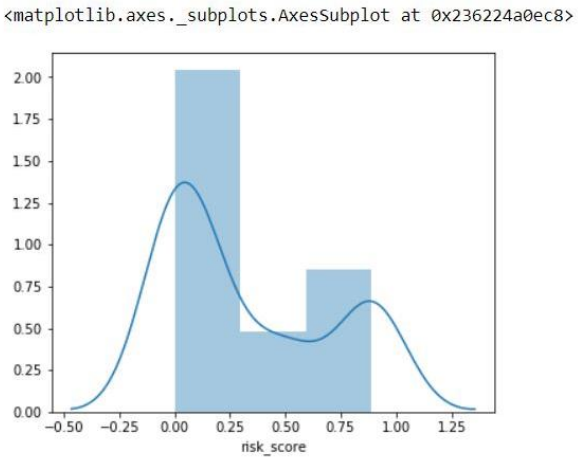



Fig 6.8

Histogram- risk_score.

### 6.1.5 Bar plot of actual Vs predicted values:

```
import matplotlib.pyplot as plt
res = res.head(20)
res.plot(kind='bar',figsize=(10,8))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

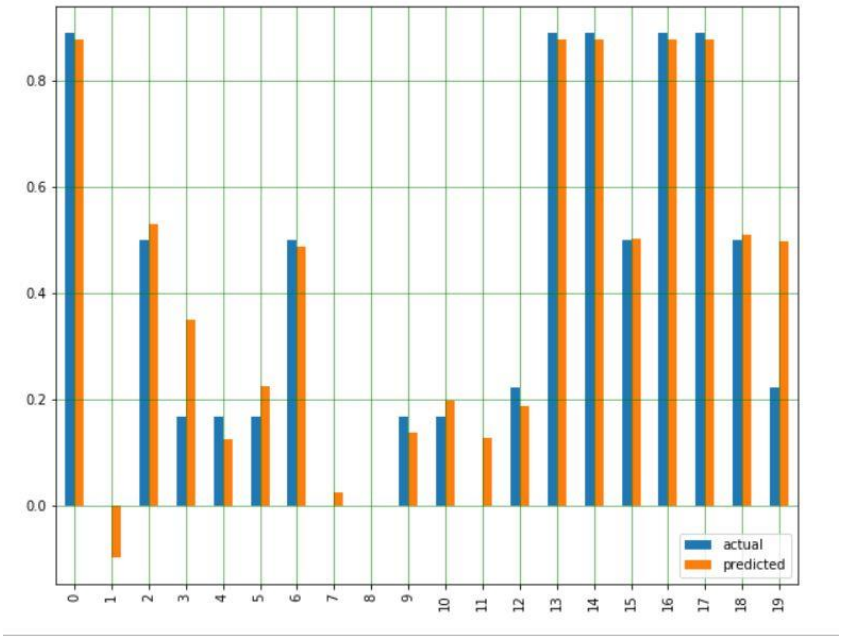



Fig 6.9

Bar graph: Actual Vs Predicted values.

- To plot the figure ‘matplotlib’ has been imported and plot() function creates the bar graph by defining the figure size as 10,8.
- As we have 100 records first 20 records have taken into consideration and represented in a bar graph format.
- Blue colour indicates the actual values and Orange colour indicates the predicted values.
- So, it shows that our model gives more exact predictions when compared with the actual values.

### 6.1.6 Original Vs Predicted visualization:

```
import matplotlib.pyplot as plt
x_ax = range(len(X_test))
plt.scatter(x_ax, Y_test, s=5, color="blue", label="original")
plt.plot(x_ax, Y_pred, lw=0.8, color="red", label="predicted")
plt.legend()
plt.show()
```

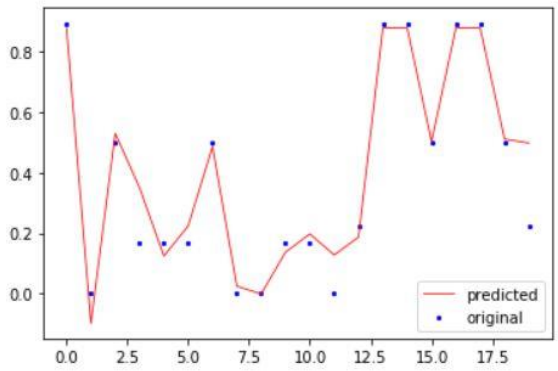



Fig 6.10

- So, the above figure shows that original values are very close to the findings i.e. the predicted values.

### *6.1.7* **Evaluating the performance of the model**:

```
import numpy as np
from sklearn import metrics
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

Fig 6.11

- In python sklearn.metrics module is installed to measure the performance of the multiple linear regression algorithm. And libraries like mean_squared_error, r2_score, etc. are imported to obtain the following performance measures.

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
print('R^2 Value:', r2_score(Y_test, Y_pred))
```

```
Mean Absolute Error: 0.05110609872115185
Mean Squared Error: 0.007318010462552507
Root Mean Squared Error: 0.08554537078388583
R^2 Value: 0.9338885094947049
```

Fig 6.12

***Performance Evaluation measures.***

- It is estimated that the model can achieve **R2 value as 0.933** which gives the accuracy of 93% i.e. the model seems to be very efficient.
- Mean Squared Error is 0.007 and Mean Absolute Error is 0.0511.
- Root mean squared error (RSME) is 0.0855 calculated by taking the average magnitude of errors. This metric is calculated by taking square root of MSE. This is used to find the efficiency of the regression technique for prediction.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Fig 6.13

- Overall, the model performance is good in predicting the risk score using multiple regression model on considering the twelve other factors. The regression equation as follows:

$$Y_{\text{predicted value}} = B_0 + B_1 * x_1 + B_2 * x_2 + \ldots + B_d * x_d$$

Fig 6.14

## 6.2 Ridge Regression:

Ridge regression is one of the sub-branches of regression techniques, that is used for analysing data that encounters a multicollinearity issue. When Multicollinearity occurs, least squares estimates becomes unbiased, and variance values increases from the original values. So, we add a degree of the bias to the regression line, that it reduces the standard error and variance values .

Following the usual notation, suppose our regression equation is written in matrix form as

$$\underline{Y} = X\underline{B} + \underline{e}$$

where $\underline{Y}$ is the dependent variable, $X$ represents the independent variables, $\underline{B}$ is the regression coefficients to be estimated, and $\underline{e}$ represents the errors are residuals.

Fig 6.15

### 6.2.1 Implementation of Ridge Regression Model:

Loading the Necessary Libraries and Splitting the Data:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
from sklearn.linear_model import Ridge
```

Fig 6.16

- First up, we are import the necessary linear_model function from the 'ScikitLearn'
- The data has been split into 80% train data and 20% test data.
- Building the Ridge Regression Model (Along with the Scores):

```
regressor = Ridge(alpha=1.0)
regressor.fit(X,Y)
```

```
Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=None, solver='auto', tol=0.001)
```

Fig 6.17

### 6.2.2. Analysing the Model Values:

```
res = regressor.score(X_train, Y_train)
print(res)
```

```
0.9756646374236978
```

```
print('intercept:', regressor.intercept_)
print('\nslope:', regressor.coef_)
```

```
intercept: [-0.58979178]

slope: [[0.24947686 0.08862744 0.08866058 0.21770697 0.07434883 0.1004104
  0.47880134 0.0506508  0.03121912 0.04814735 0.         0.01228278]]
```

Fig 6.18

- Using the regressor.score function we can find the co-efficient of determination between the independent and dependent variables in the training data.
- Then the intercept and slope of the predictor line is determined and viewed, respectively.

### 6.2.3 Prediction of Values on the Testing data:

```
Y_pred = regressor.predict(X_test)
print(type(Y_test))
print(type(Y_pred))
```

```
<class 'pandas.core.frame.DataFrame'>
<class 'numpy.ndarray'>
```

```
Y_test = Y_test.values
print(type(Y_test))
```

```
<class 'numpy.ndarray'>
```

Fig 6.19

- Now the Ridge regression model built using the training data is used for testing on the testing dataset using the predict function and the values are stored as Y_Pred.

- Now the values of the Actual test values are stored in the Y_test.

### 6.2.4 Calculating the Values of the Error and squared Errors from the Predicted and Testing datasets.

```
Y_error = Y_test - Y_pred
print(Y_error)
```

```
Y_sqr_error = Y_error * Y_error
print(Y_sqr_error)
```

Fig 6.20

```
res = pd.DataFrame({'actual':Y_test[:,0], 'predicted':Y_pred[:,0], 'error': Y_error[:,0], 'sqr_error': Y_sqr_error[:,0]})
res.head()
```

| | actual | predicted | error | sqr_error |
|---|---|---|---|---|
| 0 | 0.890 | 0.850541 | 0.039459 | 0.001557 |
| 1 | 0.000 | -0.024012 | 0.024012 | 0.000577 |
| 2 | 0.500 | 0.554140 | -0.054140 | 0.002931 |
| 3 | 0.166 | 0.219783 | -0.053783 | 0.002893 |
| 4 | 0.166 | 0.062260 | 0.103740 | 0.010762 |

Fig 6.21

- Next we calculate the Error and Squared error's in the values using the testing (Actual test values) and Predicted values obtained from the analysis.

### 6.2.5 Univariate Distribution of the Dependent variable (Risk Score).

```
import matplotlib.pyplot as plt
import seaborn as seabornInstance
plt.figure(figsize=(10,8))
plt.tight_layout()
seabornInstance.distplot(result['risk_score'])
```

Fig 6.22

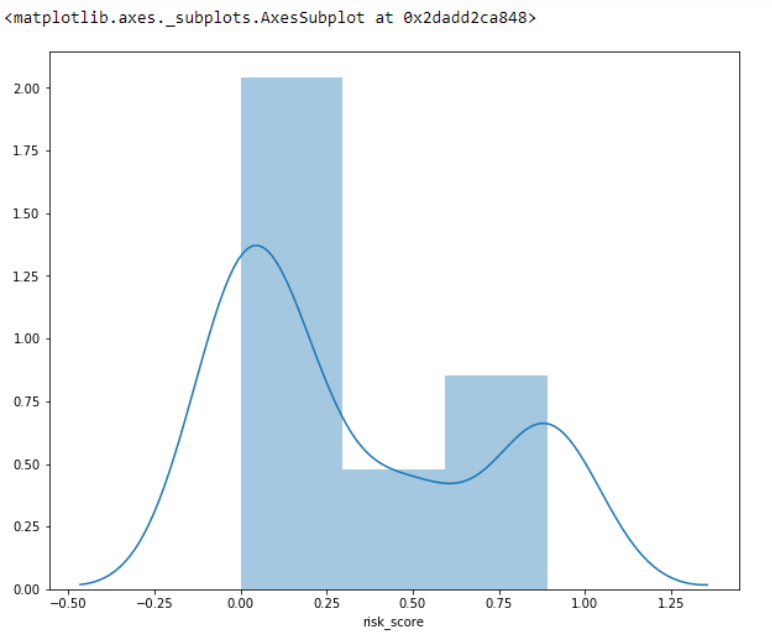



Fig 6.23

- Here we import the Matplotlib and seaborn library used for visualizing the deviation in the risk_score.

### 6.2.6 Comparing and Visualization of Predicted and Actual values:

```
df = df.head(25)
df.plot(kind='bar',figsize=(16,10))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

Fig 6.24

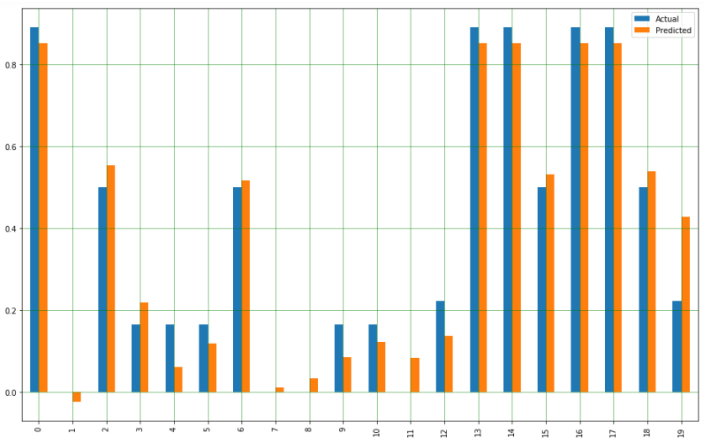



Fig 6.25

- Here we visualize and compare the predicted and Actual data values, using the error rate and squared errors formulated.

### 6.2.7 Analysing the Validation Measure of the Regression Model:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

```
print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, Y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(Y_test, Y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
print('R^2 Value:', r2_score(Y_test, Y_pred))
```

```
Mean Absolute Error: 0.05563342524496769
Mean Squared Error: 0.004812523928741262
Root Mean Squared Error: 0.06937235709373916
R^2 Value: 0.956523274782186
```

Fig 6.26

- ScikitLearn library is imported to calculate the MAE, MSE and RMSE scores. From the model we can see that the RMSE value is at 0.069 which is closer to 0 than the other models. The R^2 value is found to be at 0.95 thus making it an efficient accuracy model.

## 6.3 Lasso Regression:

- Lasso regression is a type of linear regression which uses concepts of shrinkage and regularization in the model.
- Shrinkage makes the data points to get shrunk towards the central point similar to the mean, which as a result produces a simple and a sparse model with less number of parameters. Lasso performs L1 Regularization, where the penalty equal to the absolute value of magnitude of coefficients are added so that it creates a sparse model with less coefficients and make some coefficient values to be closer to zero. Thus, creating more simpler method. Ridge regression

does not make any alterations to the coefficients or create any sparse models, thus making it difficult to interpret when compared to Ridge regression model.

- Lasso model is primarily used to avoid the problem of multicollinearity or when you want to automate the method of variable selection/elimination.
- LASSO stands for Least Absolute Shrinkage and Selection Operator.

LASSO (Least Absolute Shrinkage & Selection Operator): L1 regularization

$$\min_w \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - H\left(w, x^{(i)}\right)\right)^2 + \frac{\lambda}{2}\|w\|_1 \qquad \lambda \text{ is a regularization hyper parameter}$$

Fig 6.27

Model Building in Lasso Regression:

### 6.3.1 Loading the Required Packages and methods:

```
import pandas as pd
import numpy as np
from sklearn import model_selection

from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
```

Fig 6.28

- Here the required packages like Numpy and pandas are loaded along with Scikit-learn package when the lasso is loaded for lasso regression method to be performed and other functions like metrics, model selection are loaded to for calculating the R squared value, train and test data split, mean squared error and sqrt and for splitting the data.

### 6.3.2 Data Pre-processing and Obtaining the Required Columns:

```
target_column = result['risk_score']
predictors = list(set(list(result.columns))-set(target_column))
result[predictors] = result[predictors]/result[predictors].max()
result.describe()
```

| | FirstName | LastName | WholeName | Address | Country | PlaceOFBirth | PNumber | ExpiryDate | PlaceOFI | CountryCode | Gender | Dob | risk_score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.0 | 99.000000 | 99.000000 |
| mean | 0.802059 | 0.769846 | 0.729150 | 0.594960 | 0.585122 | 0.752946 | 0.476041 | 0.880808 | 0.680135 | 0.666667 | 1.0 | 0.875758 | 0.380907 |
| std | 0.183780 | 0.200028 | 0.227905 | 0.329527 | 0.346014 | 0.223400 | 0.313135 | 0.105634 | 0.300865 | 0.301169 | 0.0 | 0.134081 | 0.418010 |
| min | 0.500000 | 0.461538 | 0.434783 | 0.276596 | 0.210526 | 0.444444 | 0.153846 | 0.700000 | 0.333333 | 0.333333 | 1.0 | 0.600000 | 0.000000 |
| 25% | 0.615385 | 0.600000 | 0.533333 | 0.316431 | 0.285714 | 0.563492 | 0.285714 | 0.800000 | 0.333333 | 0.333333 | 1.0 | 0.800000 | 0.000000 |
| 50% | 0.800000 | 0.666667 | 0.600000 | 0.373626 | 0.375000 | 0.648649 | 0.315789 | 0.800000 | 0.666667 | 0.666667 | 1.0 | 0.900000 | 0.199041 |
| 75% | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.763158 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 |
| max | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000000 |

Fig 6.29

- Here the corresponding variables (independent variable and dependent variables) for analysis are selected and made suitable for performing the analysis.
- Then the describe function is used to view the entire data summary, along with values across various aspects as represented in the image above.

### 6.3.3 Splitting the Dataset into Training and Testing data:

```
X = result[["FirstName","LastName","WholeName","Address","Country","PlaceOFBirth","PNumber","ExpiryDate","PlaceOFI",
            "CountryCode","Gender","Dob"]]
y = result[['risk_score']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=40)
print(X_train.shape); print(X_test.shape)
```

```
(69, 12)
(30, 12)
```

Fig 6.30

- The required columns used for analysis are obtained and assigned to the corresponding variables and the dataset is split into training and testing data such that 70% of the data goes into training data and the remaining 30% goes into testing data.

### 6.3.4 Building the Lasso Regression Model:

```
model_lasso = Lasso(alpha=0.01)
model_lasso.fit(X_train, y_train)
pred_train_lasso= model_lasso.predict(X_train)
print(np.sqrt(mean_squared_error(y_train,pred_train_lasso)))
print(r2_score(y_train, pred_train_lasso))

pred_test_lasso= model_lasso.predict(X_test)
print(np.sqrt(mean_squared_error(y_test,pred_test_lasso)))
print(r2_score(y_test, pred_test_lasso))
```

```
0.09188024520654223
0.9530027177597478
0.09010797794733326
0.9484966159532163
```

Fig 6.31

- The first step, the lasso regression model is built with alpha value as 0.01 and then model is fit on the Training data with independent and dependent variables in it.
- Then Built lasso model is used for predicting on using the training data and then now using the actual training data and predicted training data, we calculate the Root mean square value and R squared values for the training data and we have found that It has an accuracy of 95% which means it's very good model.
- Now the Lasso model is used for predicting in the testing dataset and same process is repeated to calculate the Root mean square value and R Squared value for the test data, and we have found that the R squared value is 94% which means its prediction is correct for the testing dataset also and confirms the model which is built as very good model.

## 6.4 **ElasticNet Regression :**

The Elastic net regression is a regularized or an adjusted regression model. It combines both the Lasso regression technique ($L1$ norm) and Ridge regression technique ($L2$ norm). Merging them, helps to overcome the constraints faced by both Lasso and ridge regression techniques. The advantage of using this model, is that we will never know in advance about the importance of each independent variables i.e. whether to perform lasso or ridge regression model so, instead of choosing one among them the elastic net regression technique will be performed. It aids to get rid of redundant or unessential variables (Lasso) or identifies if most of the independent variables used for prediction are being useful (Ridge). It is also explained using the following equation:

$$\text{the sum of the squared residuals} + \lambda_1 \times |\text{variable}_1| + \ldots + |\text{variable}_x| + \lambda_2 \times \text{variable}_1^2 + \ldots + \text{variable}_x^2$$

Fig 6.32

From the above equation it can either be '*lambda'* or '*alpha'* if its closer to zero then its ridge regression else the same will be lasso.
So, the risk score will be predicted and analysis along with its code will be detailed below.

### 6.4.1 Loading the required libraries:

```
import pandas as pd
import numpy as np
from sklearn import model_selection

from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
```

Fig 6.33

```
target_column = result['risk_score']
predictors = list(set(list(result.columns))-set(target_column))
result[predictors] = result[predictors]/result[predictors].max()
result.describe()
```

| | FirstName | LastName | WholeName | Address | Country | PlaceOFBirth | PNumber | ExpiryDate | PlaceOFI | CountryCode | Gender | Dob | risk_sc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.0 | 99.000000 | 99.000 |
| mean | 0.802059 | 0.769846 | 0.729150 | 0.594960 | 0.585122 | 0.752946 | 0.476041 | 0.880808 | 0.680135 | 0.666667 | 1.0 | 0.875758 | 0.369 |
| std | 0.183780 | 0.200028 | 0.227905 | 0.329527 | 0.346014 | 0.223400 | 0.313135 | 0.105634 | 0.300865 | 0.301169 | 0.0 | 0.134081 | 0.412 |
| min | 0.500000 | 0.461538 | 0.434783 | 0.276596 | 0.210526 | 0.444444 | 0.153846 | 0.700000 | 0.333333 | 0.333333 | 1.0 | 0.600000 | 0.000 |
| 25% | 0.615385 | 0.600000 | 0.533333 | 0.316431 | 0.285714 | 0.563492 | 0.285714 | 0.800000 | 0.333333 | 0.333333 | 1.0 | 0.800000 | 0.000 |
| 50% | 0.800000 | 0.666667 | 0.600000 | 0.373626 | 0.375000 | 0.648649 | 0.315789 | 0.800000 | 0.666667 | 0.666667 | 1.0 | 0.900000 | 0.186 |
| 75% | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.763158 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 0.780 |
| max | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 | 1.000000 | 1.000 |

Fig 6.34

- Essential libraries have been imported and then the target column values have been fetched from the dataset.
- As detailed above, explanatory variables and the target variables mean, standard deviation, etc has been described. Data is split into test data and training data as follows.

```
X = result[["FirstName","LastName","WholeName","Address","Country","PlaceOFBirth","PNumber","ExpiryDate","PlaceOFI","Country
y = result[['risk_score']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=40)
```

Fig 6.35

- Here we separate X as the independent variables and Y as the target variable and then split them into train and test parts. We have extracted 30 percent of the dataset as the test data.

### 6.4.2 **Best Alpha Value:**

```
model_enet = ElasticNet(alpha = 0.01)
model_enet.fit(X_train, y_train)
pred_train_enet= model_enet.predict(X_train)
pred_test_enet= model_enet.predict(X_test)
print('Mean Squared Error:',np.sqrt(mean_squared_error(y_test,pred_test_enet)))
print('R^2 Value:',r2_score(y_test, pred_test_enet))
```

```
Mean Squared Error: 0.0719512622114994
R^2 Value: 0.9668458216876206
```

Fig 6.36

- Based on the alpha value, performance or accuracy of the model will differ.
- As shown above alpha is set as 0.01 and the predictions made are 96 percent accurate. On the other hand, we set alpha value as 0.0001 which increases the R square value from **0.966** to **0.979**.
- It is also stated that if the alpha value is equal or closer to zero it corresponds to be ridge regression technique and if the alpha value is set one then, the same corresponds to lasso.
- Meanwhile the Mean Squared Error value also gets decreased as shown in both the images.

```
model_enet = ElasticNet(alpha = 0.0001)
model_enet.fit(X_train, y_train)
pred_train_enet= model_enet.predict(X_train)
pred_test_enet= model_enet.predict(X_test)
print('Mean Squared Error:',np.sqrt(mean_squared_error(y_test,pred_test_enet)))
print('R^2 Value:',r2_score(y_test, pred_test_enet))
```

```
Mean Squared Error: 0.05636610669701493
R^2 Value: 0.9796531346190109
```

Fig 6.37

- This shows that model performs better with **97** percentage of accuracy i.e. R squared value when alpha is set to **0.0001**.

### 6.4.3 Visualizing original and predicted values:

```
x_ax = range(len(X_test))
plt.scatter(x_ax, y_test, s=5, color="blue")
plt.plot(x_ax, pred_test_enet, lw=0.8, color="red")
plt.legend()
plt.show()
```

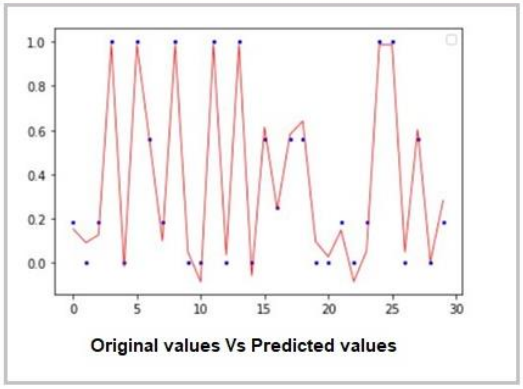



Fig 6.38

- In the above plot blue dots represents the actual values and red line represents the predicted values.
- The predicted values are very close to the original values and the same is also visualized above.

  Thus, this model finds the best combination of lasso regressions and ridge regressions characteristics by using its maximum flexibility. This model achieves 97 percentage of accuracy.

**Comparison Table:**

| Model Type | Multiple Linear | Ridge | Lasso | ElasticNet |
|---|---|---|---|---|
| MSE Value | 0.00732 | 0.005 | 0.0918 | 0.0563 |
| R^2 Value | 0.93 | 0.96 | 0.94 | 0.96 |

Table 6.1

**Result:**

From the table we can see that the Ridge Model provides more accuracy than the other models.

# Chapter 7. Conclusions and Future Work

**Conclusion:**

The research seems to conclude, that the prospects of growth in implementing AI in the field of Banking sectors seems to evolve at a rapid pace. The only way to keep in pace with this rapid growth is to include Machine Learning in the process. Thus, application built using Chatbot to develop a Sanction Checker with an inbuild OCR tool, seems to be prominently efficient. Also the evaluation of the application model is done, which proves to be highly accurate based on the Mock data generated randomly.

**Future Scope:**

The inevitability of growth in Artificial Intelligence seems eminent in Banking sector. Although, it may seem crucial for all banking firms to adopt such revolutionary AML technologies, it is also equally important for financial institutions to impose regulation on such practises. With the threat of committing financial crimes increases exponentially with the recent natural calamities, it is high time that Automation of Banking happens sooner than later. But for the banking firms to adopt such automated methods, the scope to improve seems ample but the intent to be adaptable seems to be feign most times if not always.

# Chapter 8. Bibliography

1.Sharon,C.,Elizabeth,J.,Tamar,K.,(2018),"Anti-Money Laundering Enforcement: The Rise of Individual Liability for Compliance Professionals".

2.Alexander.W,(2019),"Europe's Finance Chiefs to Call for Anti-Money Laundering Agency".

3.JC,.S,David,C.,(2019),"Corruption and Anti-Money-Laundering Systems: Putting a Luxury Good to Work"

4.David,F.,Kevin,Z.,(2004)," Bank Regulators Bring Enforcement Actions for Alleged Shortcomings in AML Policies and Procedures"

5.Penny,C.,(2018),"AI as new tool in banks' crime-fighting bag?"

6.Penny,C.,(2019),"Is regulators' green light on AML tech a game changer?"

7.Jeff,S.,Piotr,K.,(2018),"Monitoring Money-Laundering Risk with Machine Learning"

8.Nathan,D.,(2018),"Fighting financial crime without excluding the underbanked"

9. Accenture Leveraging-Machine-learning, (Available at:https://www.accenture.com/_acnmedia/pdf-61/accenture-leveraging-machine-learning-anti-money-laundering-transaction-monitoring.pdf),(Accesssed 4 June 2020).

11.BobsGuide,("Available at:/guide/news/2019/Nov/1/ai-vs-money-laundering-who-wins/") AI vs money laundering. Who wins?,(Accessed:4 June 2020)

12"How to trust the machine: using AI to combat money laundering",Available at: file:///C:/Users/akile/Zotero/storage/57Q65X6I/2019%20-

%20Application%20of%20Machine%20Learning%20in%20AML%20Transaction.pdf(Accessed: 4 June 2020)

13.”Application of Machine Learning in AML Transaction Filtering”,(Accessed: 4 June 2020)

14.”Fighting Financial Crime with AI”, (Available at:"https://www.ibm.com/downloads/cas/WKLQKD3W",(Accessed: 5 June 2020)

15. “Simple KYC & Due Diligence Check | CDD Made Easy”,(Available at:https://www.ey.com/en_ie/trust/how-to-trust-the-machine--using-ai-to-combat-money-laundering),(Accessed: 5 June 2020)

16. “Strengthening AML protection through AI”,(Available at: file:///C:/Users/akile/Zotero/storage/97IZGJW9/strengthening-aml-protection-through-ai.html#.XuPsOUVKhPY), (Accessed: 12 June 2020)

17. Kerry,H.,(2018),”How The Coronavirus Is Impacting Small Business Owners”
.
18. Maria, W.,(2016),”Coronavirus: Why is it so hard to aid small businesses hurt by a disaster?”

19.”DHHAN, W. (2017) ELASTIC NET FOR SINGLE INDEX SUPPORT VECTOR REGRESSION MODEL. Economic Computation and Economic Cybernetics Studies and Research, Issue”.

20.Eva, O. & Oskar, O. (2011) “The Simple Exponential Smoothing Model”
.
21.Jama,. (2020) “Statistic guide for students and researchers with SPSS illustration. Chicago: Abdiasis”.

22.Madhuri, C.R., Madhuri, , G, & Pujitha ,. (2019) “House Price Prediction Using Regression Techniques: A Comparative Study. IEEE 6th International Conference on smart structures and systems ICSSS “.

23.Darrin, D., 2020. “Elastic Net Regression In Python. [online] educational research techniques.” Available at: <https://educationalresearchtechniques.com/2018/12/24/elastic-net-regression-in-python/> [Accessed 12 June 2020].

24.Abdullah,A.,Alma,I.,Rahman,A.,(2010)”Programming challenges of Chatbot: Current and Future Prospective”

25.Ayush,G.,(2020),”Development of Chatbot Using Deep NLP and Python”

26.Ibrahim,A.,(2019),”Create a chatbot – Using Python, Recurrent Neural Networks and TensorFlow”

27.Ranjit,S,(2020)”Google Cloud Platform Administration: Design highly available, scalable, and secure cloud solutions on GCP”

28. PatrickV.,Thijs,K., Vasilios,A., Mircea,.L,(2018),”A Low-Effort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services”

29. Jamshed, M.Maira,S.,Rizwan,A.,(2018),”Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)”

30.Somayajulu,S. Ehud,R.,(2020),”Acquiring Correct Knowledge for Natural Language Generation”

31.Sebastian,E.,(2020),”Design and Prototypical Implementation of an Open Source and Smart Contract-based Know Your Customer (KYC) Platform”

32. Jolanta Ciak,R.,(2016),"Politically Exposed Persons Approach in EU Financial Institutions: Legal Frames and Business Practice Divergence (Polish Case)"