AI-Statistical Machine Learning Approaches to Liver Disease Prediction

Team ID: PNT2022TMID48272

Faculty Mentor:

D.**Pradhiba**

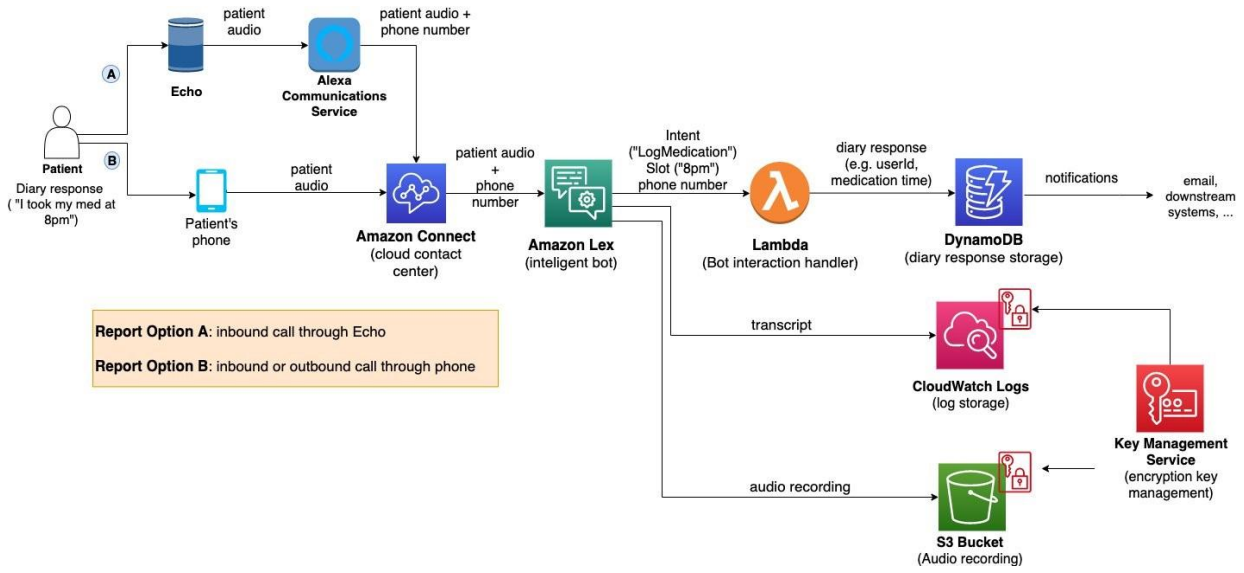
Team Leader: G.lydia
Team Member: R.priya
Team Member: U.lavanya sri
Team Member: G.nagalakshmi

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:



In clinical research studies conducted by the Pharmaceutical industry, electronic clinical outcome assessments (eCOAs) help clinical researchers

better understand the effectiveness of the research treatment intervention for research patients. One type of eCOA measure is **electronic patient-reported outcomes (ePROs):** Health outcome data reported directly by the research patient, typically collected on handheld mobile devices, tablets, or computers. Another eCOA measure is the eDiary: Electronic collection of subjective data from a research patient with automated data entries on a handheld mobile device, tablet, or computer. Studies have shown that compared to paper diaries, eDiaries are more effective at improving patient compliance with the research protocol and therefore generate more accurate data. A popular way to collect diary data has been through mobile phone apps due to the ubiquity of smart phones. Although many patients find it convenient to receive reminders and record information using their smartphones, this approach does not work for everyone. Some patients may have medical conditions such as vision impairments or motion impairments (e.g. Parkinson's or ALS), making it difficult to enter data on a phone screen. In other clinical trials, researchers may work with patients without sufficient literacy to read the question prompts. For these patients, a system that allows outcome reporting through voice offers a great alternative. Building blocks from AWS such as Amazon Lex and Amazon Connect enable clinical researchers to quickly stand up a **hands-free voice interface for patient outcome reporting**. By leveraging capabilities built into AWS services, you can build natural, conversational patient experiences using an automated voice reporting system.

The AWS Envision Engineering America's Prototyping (EE) team co-builds working prototypes to demystify technology and ignite customers' innovation engines so that they are empowered to invent on behalf of their customers. We recently consulted with PRA Health Sciences, one of the world's leading global contract research organizations (CRO). Dr. Isaac Rodriguez-Chavez, SVP Head of Global Center of Excellence Decentralized Clinical Trials at PRA, commented, "To make clinical trials more inclusive and less burdensome, PRA is working to incorporate voice enabled technology into research protocols and exploring how the Alexa device can interact with PRA's Mobile Health Platform to collect quality, regulatory

grade clinical trial data. This collaboration supports PRA's wider initiatives to enable and improve decentralized clinical trials." The following image shows a high-level architecture and data flow diagram of the voice-enabled patient diary sample implementation. Note that the interaction is two-way in nature:

The application supports three different ways for patients to reach the patient dairy voice reporting interface:

- Making an inbound call through their smart phone or landline
- Making an inbound call through a smart speaker such as Amazon Echo
- Receiving an outbound call on their smart phone or landline, initiated by an automated patient outreach scheduler that uses the StartOutboundContact API of Amazon Connect (the outreach scheduler is out of the scope of this blog)

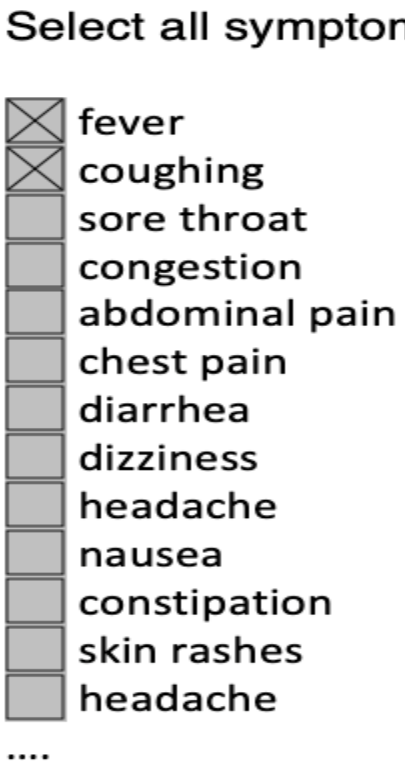
Regardless of which method the patient uses to reach the reporting application, the patient call is handled by Amazon Connect. In this application, the contact flow first greets the patient, then verifies their identity. If successfully authenticated, the flow then checks which questionnaires the patient needs to complete, and collects responses for each questionnaire accordingly.

To support natural language interactions and interpret the patient's response, the Amazon Connect contact flow delegates each questionnaire to a corresponding Amazon Lex bot. The Amazon Lex service allows you to build conversational interfaces using voice and text. It provides developers the same deep-learning based, conversational AI building blocks that power Amazon Alexa: automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text. You can build a conversational interface for your application in minutes by supplying a few example phrases for each intent you want to support.

After the Amazon Lex service interprets the patient's response and parse it into semi-structured data (e.g. medicationTime='20:00'), it invokes an **AWS Lambda function** where you can define custom business logic to react to this information, such as storing in a clinical research database or an EHR system or notifying the principal investigator or the patient's caregiver if the patient has missed their medication. In the prototype application, the reported information is stored in a set of Amazon DynamoDB tables, and notifications to caretaker/providers are implemented using Amazon Pinpoint.

Voice user interface design considerations

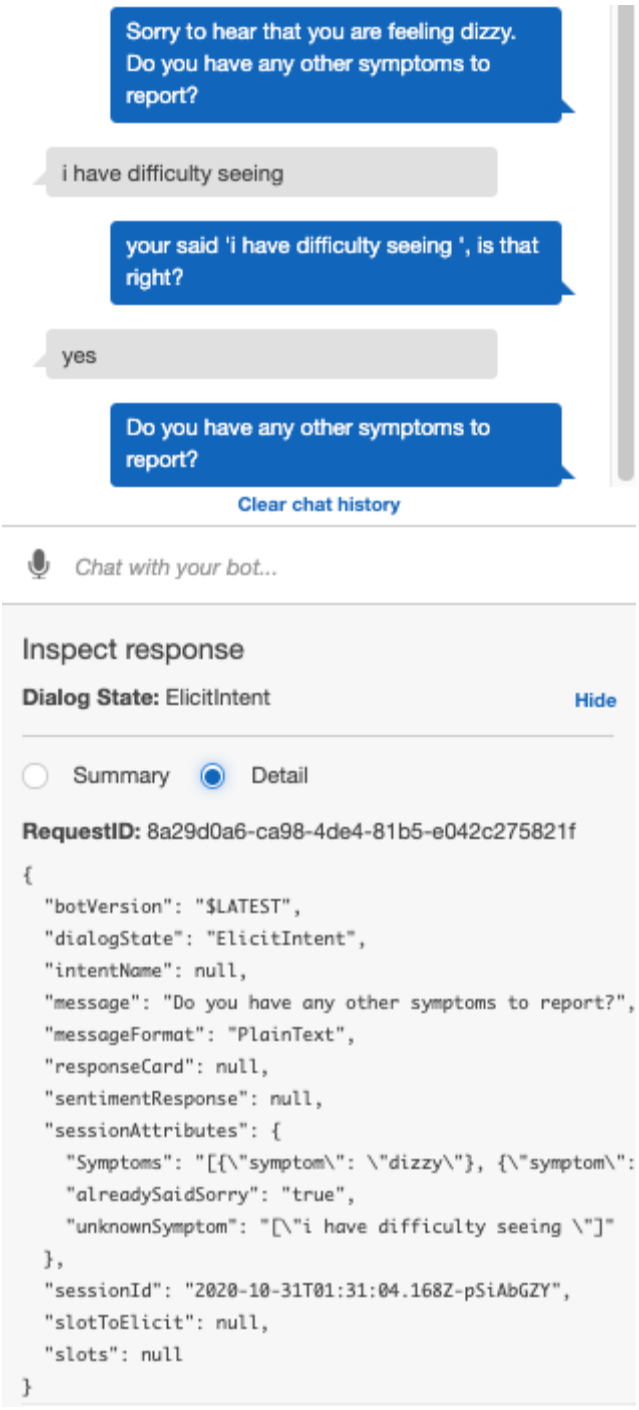
Designing voice user interfaces (VUIs) require a very different approach compared to building graphical user interfaces (GUIs). For example, in a typical web or mobile app, a symptom collection questionnaire is often comprised of a form with a long list of symptoms for which the patient can tick off checkboxes:



Directly translating such a checklist into a voice interface by reading through each option and asking the patient yes or no would surely wear out the patience of any potential users quickly. A best practice in building a natural and user-centric VUI is allowing users to **speak in their own words.** By leveraging the NLU capabilities provided by Amazon Lex, we can build a VUI so that patients can respond to the question "*what symptoms do you have?"* in the same way they would talk to a physician, such as "*I'm having a headache*."

To build an Amazon Lex bot, you define distinct **intents** that each represents a meaning the user tries to convey. Then you provide "sample utterances" for each intent, which will be used as training data by Lex to build machine learning models to recognize the correct intent. It's important to provide a wide range of sample utterances when defining each intent as there may be a variety of ways users could respond to a given question. **Testing with users with diverse backgrounds** can help identify ways of expression you might not consider otherwise.

However, even with a thorough set of sample utterances in place, the bot could still fail to interpret the user from time to time. You should have a plan for **gracefully handling errors**. One way to do this is by using the built-in **fallback intent** from Lex. When the bot could not map the user's input to any of the custom intents defined in the bot, and a fallback intent is defined, the bot custom logic you define to handle a fallback intent. In our sample application, we programmed the fallback intent for the symptom collection bot to confirm the user's input by simply repeating it back to them. If the user says "yes", the system can simply store the entire utterance into the reporting database as a symptom with an "unknown" flag and move on with the conversation. This allows alerting a system admin to review the unknown input after the fact without interrupting the current user reporting flow.



For more tips and design patterns on building great voice user interfaces (VUIs), refer to the VUI design guide from the Amazon Alexa team.

Security and compliance considerations:

Collecting and storing patient information may be subject to a variety of regulatory and compliance standards depending on the use case, such as **FDA Title 21 CFR Part 11**, HIPAA, HITRUST, and SOC, as well as other clinical trial, research confidentiality, and record keeping obligations. While the AWS security home page has a lot of resources for guidance and best practices for securing your AWS workloads in general, in this section, I will highlight a few security and compliance considerations specific to the voice patient reporting application and how to leverage AWS services and features to address them.

Patient authentication:

In a typical web or mobile application, developers often implement two-factor authentication by sending a verification code to the user's email or through a SMS text. Although a similar approach may be implemented in a voice reporting workflow, this functionality may be challenging for many target users of the reporting system to complete: patients with vision or motion impairments who prefer a "hands-free" experience.

In the sample application, user authentication is implemented by verifying two pieces of information from the user: the calling phone number (through automatic number identification, or ANI in short) and a security question they can answer through voice (e.g. asking for their zip code). This is done by passing the ANI phone number to an Amazon Lex bot to look up the patient's profile and verify their answers to the security question. When used in research, ways in which to manage this information to prevent the revelation of individual is discussed below.

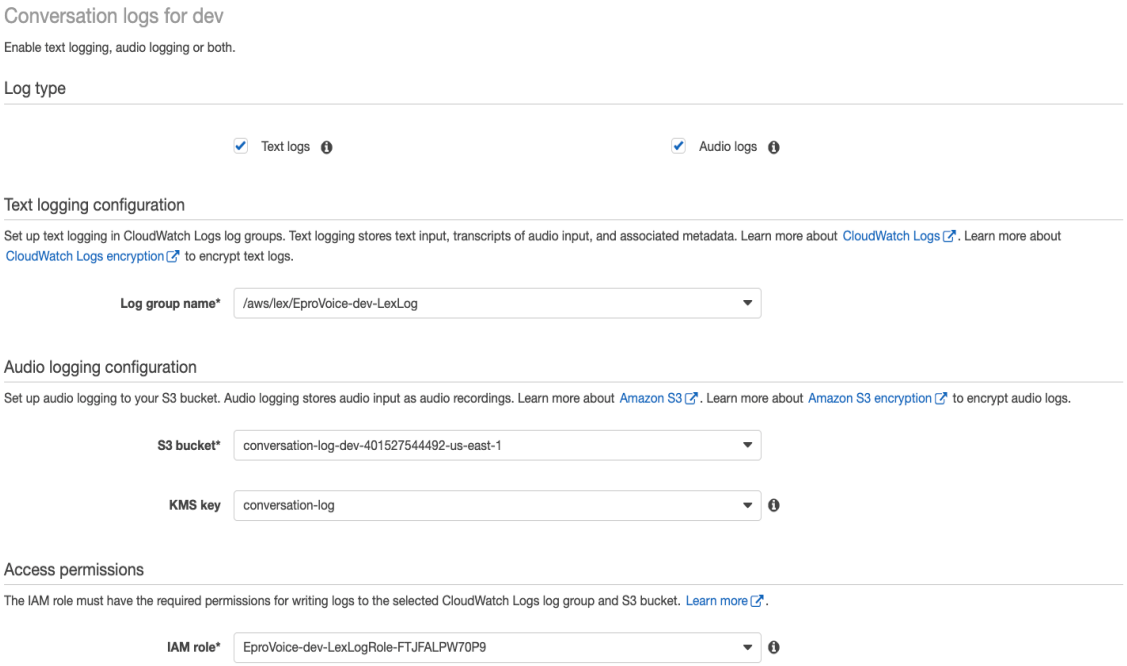
Besides the above-mentioned methods, one additional factor for verifying user identity is through voice biometrics. Although not part of the sample application, there are several voice biometric offerings provided by Amazon Connect technology partners that can be integrated into the application or consider the Amazon Connect Voice ID feature.

Auditability and traceability:

At AWS, security is job zero. We can leverage the features of AWS services—relying on the built-in security—to make every request traceable, auditable, and secure. The FDA Title 21 CFR Part 11, which regulates the keeping of electronic records for clinical trials, requires controls in place to "ensure the **authenticity**, **integrity**, and, when appropriate, the **confidentiality** of electronic records". To provide a secure audit trial of patient records and operator actions, several AWS services and features can be leveraged in combination to make every request to the system auditable and traceable for the architecture outlined above.

A useful feature for Amazon Lex that enables auditability of patient input records and troubleshooting is Lex conversation logs. It supports logging both the text transcript and capturing the audio recording of each user utterance as a separate audio file, supporting identity protection as noted in the following image. The text logs are stored in Amazon CloudWatch Logs, while the audio logs are stored in Amazon S3.

The following image shows the Lex console UI enabling the two types of logs respectively. Note that both the audio and text logs can be configured to use customer managed Amazon KMS encryption keys for additional security compliance.



In addition to Amazon Lex conversation logs, other AWS service and features can add auditability to additional parts of the voice patient diary architecture, all with identity protection features noted:

- Amazon Connect contact flow logs (for capturing the steps that each contact goes through)
- CloudWatch Logs for AWS Lambda Functions (for logging details of the user input processing code)
- AWS CloudTrail (for capturing infrastructure configuration changes)
- Amazon DynamoDB Streams (for capturing changes made to DynamoDB tables)

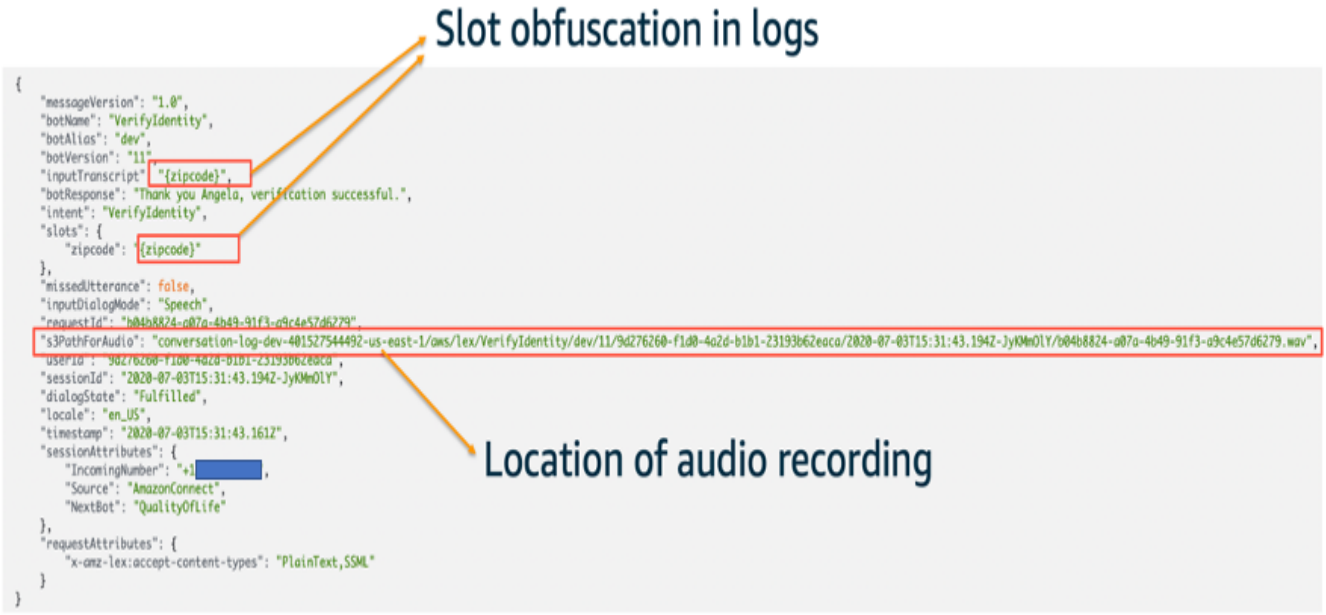
PII and PHI data protection:

As the AWS shared responsibility model for security describes, while the underlying service infrastructure is secured by AWS and verified by a number of compliance programs, the clinical research sponsor or CRO is responsible for securing the application built on AWS. For example, as Amazon Connect invokes AWS Lambda and Amazon Lex on the CRO/sponsor's behalf, all data exchanged between these services is

encrypted in transit by AWS (see Connect and Lex documentation). However, as the CRO/sponsor is in control of where to send and store the patient reported data received by the AWS Lambda function, the CRO/sponsor is responsible for securing its storage and transmission downstream. CROs/sponsors should refer to the HIPAA on AWS, GxP on AWS, and HITRUST on AWS pages for considerations for specific compliance programs that may apply to each use case.

When working with PII (personal information) and PHI (health information) data, CROs and sponsors need to be intentional about where PII data is stored and sanitize sensitive information so they are not unintentionally exposed in logs used for operational purposes. There are a few different types of logs to consider when using Amazon Lex and Amazon Connect.

For Amazon Lex, we discussed the Lex conversation logs feature in the previous section, which adds auditability to Lex bots in your application. If you enable conversation text logs in Lex, one helpful feature for protecting sensitive PII is Slot Obfuscation. When enabled, Amazon Lex replaces the value of the slot with the slot name in conversation logs. You can see this obfuscation in action in the example CloudWatch Logs entry:



These conversation logs also contain values you store in Lex session attributes. If you store any sensitive PII information in session attributes and do not want them to appear as clear text in logs, you need to encrypt or otherwise obfuscate these values before sending them to the Lex service.

In Amazon Connect, information about each contact are automatically captured in contact trace records (CTR). Data in CTRs powers metrics and reports that you can review to gain insights into your Connect instance. When building your contact flow in Connect, you might use contact attributes to store information about the customer to personalize the experience (similar to the session attribute concept in Amazon Lex discussed previously). For example, you may set the patient's name as a contact attribute after looking it up in an Amazon Lex bot and then use it to address the patient when playing a question prompt. If you do not want this information to be stored in the CTR record, you can set its value to an empty string to "erase" it before the ending the contact flow.

Summary:

In this post, we discussed the architecture and key considerations for building an automated yet human-centric, secure, and auditable application for research patient outcome reporting through a conversational voice interface. For example, this enhances the ability of clinical research sponsors and CROs to understand patients' experiences with research interventions and encourages compliance by making reporting easier and more accessible to more patients.

In part two of this blog post, we will walk through key components of the sample application and how to set it up in your account.

Solution Architecture information:

Today's health care is very important aspect for every human, so there is a need to provide medical services that are easily available to everyone. In this paper, the main focus is to predict the liver disease based on a software engineering approach using classification and feature selection technique. The implementation of proposed work is done on Indian Liver Patient Dataset (ILPD) from the University of California, Irvine database. The different attributes like age, direct bilirubin, gender, total bilirubin, Alkphos, sgpt, albumin, globulin ratio and sgot etc, of the liver patient dataset, are used to predict the liver diseases risk level. The various classification algorithms such as Logistic Regression, SMO, Random Forest algorithm, Naive Bayes, J48 and k-nearest neighbor (IBk) are implemented on the Liver Patient dataset to find the accuracy. The comparison different classifier results are done of feature selection and without using feature selection technique. The development of intelligent liver disease prediction

software (ILDPS) is done by using feature selection and classification prediction techniques based on software engineering model. This disease contains many conditions like inflammation (hepatitis B, C) from infectious, non-infectious causes (chemical or autoimmune hepatitis), Tumors, malignant and scarring of the liver (Cirrhosis) and Metabolic Disorders. In this paper, six classification algorithms J-48, Random Forest, Logistic Regression, SMO (Support Vector Machine), IBk (k nearest Neighbor), Logistic Regression, Naive Bayes have been considered for implementation and comparing their results based on the ILPD (Indian Liver Patient Dataset) Software's developed for the healthcare plays a crucial role in delivering efficient services to the physicians and

ultimately better treatment can be offered to the patients. An efficient healthcare software can assist in several activities like forecasting of the diseases based on the historical data of some another patient, image processing of medical images, a data warehouse for management of the whole institution etc. Proposed work focuses on the development of the software that will help in the prediction of the level diseases based upon the various symptoms. The development stage of the given software includes continuous interaction with the physicians so that more accurate results can be generated. The data may contain redundant and irrelevant attributes, there is a need to remove these attributes withoutdecreasing the accuracy using a feature selection technique. In the present work, the WEKA tool is used for the implementation of a feature selection technique. There is a number of feature selection methods available in the WEKA environment. Feature selection can provide us with several benefits like reducing lifting, reducing training time and improving accuracy etc. In Weka tool, the feature evaluator and search method were used to perform feature selection on any dataset. In our proposed work the Correlation-based Feature Selection Subset Evaluator was used as Feature evaluator and Greedy Stepwise used as search method Based on above techniques these 5 features/attributes are select .

In their research work authors have worked on doing an analysis of the data related to Liver Disorder with the help of Naive Bayes, Decision Table, and J48. However, attributes like case history of the patient, diabetes, smoking, obesity, alcohol intake, smoking etc were used. Based upon the given database it has concluded that male people are having more liver disorder than the females. Age group of 35-65 is mostly affected and out of these 26% people are having the disorder because of alcohol, smoking contributed to 22% of people, obesity, and diabetic of 4 & 5 percent respectively. A. Gulia et al. in their proposed work researchers have done classification of the liver patient data using the algorithms like Bayesian Network, Support Vector Machine, J48, Multi-Layer Perceptron and Random Forest. The data from the UCI repository which is afforded by Center of Machine Learning and Intelligent Systems has used. After completion of their three-phase analysis, the Random Forest Algorithm is the bestone with an accuracy of 71.87% has been concluded. Y. Kumar et al. [12] in their proposed work researchers have used Rule-Based Classification Model (RBCM) for the prediction of liver diseases. Without the rule-based classification the efficiency of all the common algorithm decreases was analyzed. In their proposed work 20 rules were used for the classification of liver diseases. The decision tree-based algorithm gives the best performance using rule-based classification and accordinglyits accuracy decreases when rule-based is not used. M. Pasha et al. work on the dataset from the UCI repository is used which is having 583 instances, the meta- learning algorithms like Grading, logit boost, Adaboost, and Bagging were used. The comparisons of the algorithms based upon the amount of correct and incorrect classifications and time of execution have done. After doing detailed analysis the grading is the best algorithm in terms of accuracy and execution time have been concluded. M. Abdar et al. [14] in their research work focuses on the early prediction ofliver diseaseusing Multilayer Perceptron Algorithm (MLPNN) which uses (CART) (classification and regression tree, (CHAID) Chi-square

Automatic interaction detector, See5(C5.0). Their dataset is from UCI repository of the University of California, Irvine relevant to Indian Liver Patient Dataset (ILPD). From their results, it can be concluded that MLPNNB-CHAID is the best algorithm with an innovative accuracy of 14.57%. The 70% of the data as a training data and rest of the 30% for the testing stage were used. Author EI-Shafeiy at al. [15] in their research work focuses on electronic health records, metabolomics analyses are some of the digital information related to the health and with the passage of time, the shape of Big Data has been taken. In the present work dataset having 23 attributes of 7000 patients with 5295 as male and rest as female is used. Their proposed work make use of Boosted C5.0, Support Vector Machine, Naïve Bayes (NB) along with the feature selection. Vijayarani et al. [16] in their research paper classification algorithms are used for the prediction of liver diseases. Famous algorithms like Naïve Bayes and Support Vector Machine (SVM) are used in the proposed work. The dataset from the UCI repository and it is having fields like Gender, Sgot, ALB, ALP, DB etchave taken. Based upon their present work SVM is best in terms of accuracy and Naïve Bayes is good in terms of execution time. Soegijoko et al. in their proposed work have developed e-Health system for the improvement of community health care system. The use of various Open source software elements like Gammu as SMS engine, MySQL for database handling and Apache as a web server are used. The authors proposed system classifies the mother and child care activities as automatic message generation for the first-trimester pregnancy, second and third -trimester pregnancy, post-partum (after delivery), 40 days after delivery etc. Implementation issues like increasing electricity power, infrastructure preparations, health providers, education and training, technical hardware/software problems etc has highlighted. Their initiative for the paperless prescription system helps in reducing the paperusage is very effective. Skevoulis et al. in their proposed work engineered a software which calculates the Coronary Heart disease (CHD) risk factor. It is a joint effort between the Pfizer Inc's and Seidenberg School of Computer Scienceat Pace University (SCSIS). Their

proposed work provides the choice of selecting three CHD risk algorithms, a customized health report and a progress chart. Their proposed work makes use of client-server-based model and the content is dynamically generated using ASP.NET programing language and C#, Microsoft SQL server 2000 and Crystal reports for generating the health reports, ChartFX for representing progress chart. Their proposed work also makes use of JAVA and JavaScript for displaying CHD results. Weber-Jahnke in their proposed work discusses various challenges related to Software engineeringin Health Care (SEHC). Their proposed work requires the community to adopt knowledge translation (KT) for better resultgeneration by the software systems. Their recommendation of using the KT is based upon the survey which highlights that number of software engineering tools, methods, processes that works perfectly for other domains cannot be easily transferred to the healthcare applications. Patent Dataset The total number of 583 instances/rows with eleven distinct attributes were composed from the (ILPD) Indian Liver Patient Dataset from the UCI Repository [17] to solve the purpose of this paper. The attribute “class” described as the disease with value “1” mean a person having liver disease present and “2” represent liver disease not present. Table 1 shows the description of attributes, values of liver disease database. The database having 416 with a patient having liver diseases and 167 are non-liver diseases instances .