"PREDICTING RISK FOR NEW INSURANCE USING CUSTOMER INFORMATION"

A Major Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Engineering in *Computer Science Engineering*

Submitted by:

Durgesh Sharma (0827CS181072)
Harshala Gaikwad (0827CS181087)
Kartavya Verma (0827CS181103)

Guided by:

Prof. Narendra Pal Singh Rathore
Associate Professor
CSE, AITR, Indore

Acropolis Institute of Technology & Research, Indore

Jan - June 2022

EXAMINER APPROVAL

The Project entitled ***"Predicting Risk for New Insurance using Customer Information"*** submitted by **Durgesh Sharma (0827CS181072), Harshala Gaikwad(0827CS181087), Kartavya Verma (0827CS181103)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Engineering degree in Computer Science Engineering*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled ***"Predicting Risk for new insurance using customer information"*** submitted by Durgesh Sharma (0827CS181072), Harshala Gaikwad (0827CS181087) and Kartavya Verma (0827CS181103) is a satisfactory account of the bonafide work done under the supervision of ***Prof. Narendra Pal Singh Rathore***, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that project entitled ***"Predicting Risk for New Insurance Using Customer Information"*** has developed by us under the supervision of ***Prof. Narendra Pal Singh Rathore***. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Durgesh Sharma (0827CS181072)
Harshala Gaikwad (0827CS181087)
Kartavya Verma (0827CS181103)

Acknowledgement

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Prof. Narendra Pal Singh Rathore,** Associate Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof. Narendra Pal Singh Rathore,** Associate Professor, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr. S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Durgesh Sharma (0827CS181072), Harshala Gaikwad(0827CS181087), Kartavya Verma(0827CS181103)

Executive Summary

Predicting Risk for New Insurance Using Customer Information

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Computer Science Engineering branch under the sagacious guidance and vigilant supervision of **Prof. Narendra Pal Singh Rathore**.

The project is based on Supervised Machine Learning Technique. In this project we train our dataset (containing the data of previous policy holder along with their risk level) using Random Forest Algorithm. The purpose of this project is to predict risk level of new customer in order to accept or reject the insurance policy based on the information they provided.

Key words: Supervised ML, Random Forest

“Where the vision is one

year, cultivate flowers;

Where the vision is ten years,

cultivate trees;

Where the vision is eternity,

cultivate people.”

- Oriental Saying

List of Figures

List of Tables

List of Abbreviations

Abbr1: R- CNN- Regional based Covolutional Neural Networks

Abbr2: COCO – Common Objects in context

Abbr3: OpenCV- Open Source Computer Vision

Abbr4: JSON- Java Script Object Notation

Abbr5: CIF- Count In Frame

Abbr6: GPU- Graphical Processing Unit

Abbr7: YOLO- You Only Look Once

Table of Contents

Chapter 1. Introduction

Introduction

Risk management is important for insurance industry to identify new authenticated customers. Therefore, risk prediction of new insurance is a crucial factor in insurance business to classify the applicants. Analyzing the profile of individual applicant manually may take a very long time. By applying predictive modelling techniques may automatically classify risk level based on past data more quickly and accurately in less time and less labor.

Our project uses the dataset containing past customer information including age, height, bmi, family history, insurance history etc. along with the risk level. Hence here Supervised Machine Learning Algorithm is used to predict the risk level of different applicants. We apply Random Forest Algorithm to predict the risk level of new customer automatically.

1.1 Overview

The project is based on Supervised Machine Learning Technique. In this project we collect dataset of insurance company containing the data of previous policy holder along with their risk level from kaggle and trained our dataset using Random Forest Algorithm. Our project take the input information provided by customer and predict the different risk level based on the information provided.

The purpose of this project is to predict risk level of new customer in order to accept or reject the insurance policy based on the information they provided.

1.2 Background and Motivation

Instead of treating all customers equally, managers have come to understand that it is more effective to develop customer-specific strategies. As a result, companies are now adopting customer relationship management (CRM). CRM means that companies manage relationships with individual customers with the aid of customer databases and interactive mass customization technologies. The adoption of CRM has been enhanced by recent developments in Information and Communication Technology. By using customer information contained in databases, companies can invest in the customers that are potentially valuable for the company, but also minimize their investments in non-valuable customers. Researches in this genre do not target at analyzing particular insurance types but discussing issues of risk classification based on dichotomy instead studying cases of both perfect information and information asymmetry and of both competitive and non-competitive markets.

Risk Prediction in insurance company becomes a necessity for the betterment of company. Manual prediction may take a very long time therefore by doing this automatically may not only give better performance and reliability but also can help in predicting risk in less time and less efforts. This can be done using Predictive Modelling approach.

1.3 Problem Statement and Objectives

Risk management is an important factor in any insurance industry. Insurers consider every available quantifiable factor to develop profiles of high and low insurance risk. Level of risk will be able to determine the expected insurance premiums.

Generally, insurance policies involving factors with greater risk of claims are charged at a higher rate. For this insurer collect a vast amount of

information about policy holders. Calculating risk manually may take very long time. Therefore, calculating it using Machine Learning model may increase accuracy, performance and risk can be predicted automatically. Thus, the system implemented has the following objectives:

1. **Objective 1:** To classify applicant risk level based on their information and past available data by automation with the help of machine learning, and thus reduce manual efforts and increase performance.
2. **Objective 2:** To find most appropriate model that can classify applicant to different risk level with best accuracy and in less execution time.

1.4 Scope of the Project

- Predictive modeling utilizing learning algorithms can give the eminent contrast in the manner which business is done as compared with the traditional strategies.
- With information expository arrangements, the work should be possible quicker and with better outcomes.
- Future work identifies with the more top to bottom investigation of the issue and new techniques to manage specific systems.
    - Client division is the division of the informational index into gatherings with comparable credits can be executed to section the candidates into gatherings with comparable qualities dependent on the characteristics present in the data-set.
    - For instance, comparable work history, protection history and restorative history.
    - Following the gathering of the candidates, prescient models can be executed to add to an alternate information digging approach for the disaster protection client informational collection.

1.5 Team Organization

- **Durgesh Sharma:**

  Gathered ideas and worked on project selection. I developed a detailed project plan to track progress of the project activities, surveyed the various resources and developed presentations and other related documents. I coded the project at initial stage. I have contributed in documentation, reports, research paper and other design related diagrams (UML diagrams). Posters and animation videos were developed by me.

  Attended and contributed to all sessions, meetings and activities as required.

- **Harshala Gaikwad:**

  I have selected the project topic and done preliminary investigation, I studied about the topic, its scope and surveyed various research papers related to the importance of risk prediction in insurance companies.

  I have also collected the data-set from kaggle that fulfilled the requirement of our project and applied various predictive modelling techniques to find the best suited model. I helped in the deployment part of our project which is done by using Django Framework. Along with this I also worked on the designing part of the project (designs the UML diagram), project report, ppt presentation and research paper.

- **Kartavya Verma:**

  I investigated and found the right technology and studied in deep about it. For the implementation of the project, I build the model for the project. Implementation logic for the project objective and coding of internal functionalities is also done by me.

1.6 Report Structure

The project ***Predicting Risk for new insurance using customer information*** is primarily concerned with the **risk prediction of new customer using predictive modelling** and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. Review of Literature

Review of Literature

Risk assessment is a crucial element in the life insurance business to classify the applicants. Companies perform underwriting process to make decisions on applications and to price policies accordingly. With the increase in the amount of data and advances in data analytics, the underwriting process can be automated for faster processing of applications. This research aims at providing solutions to enhance risk assessment among life insurance firms using predictive analytics. The real world dataset with over hundred attributes (anonymized) has been used to conduct the analysis.

2.1 Preliminary Investigation

2.1.1 Current System

- In current system different ways of data mining and multivariate analytic techniques was described that can be used to improve decision making processes in such functions as life insurance underwriting and marketing.

- In this system, several machine learning techniques to analysis the insurance claims efficiently and compare their performances using various metrics.

2.2 Limitations of Current System

The limitations of these are as follows :

- Risk factor and mean standard evaluation is not done.
- Security risk with a lot of confidential data of the user.
- User data collection was not up-to the mark.
- Data used was limited which causes in unpredictability of the result.
- Data security was not up-to the mark.

2.3 Requirement Identification and Analysis for Project

Significant work has been done in the field of prediction of risk in life insurance. ; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under :

Over the years, life insurance companies have been attempting to sell their products efficiently, and it is known that before an application is accepted by the life insurance company, a series of tasks must be undertaken during the underwriting process [1].

According to [2], underwriting involves gathering extensive information about the applicant, which can be a lengthy process. The applicants usually undergo several medical tests and need to submit all the relevant documents to the insurance agent. Then, the underwriter assesses the risk profile of the customer and evaluates if the application needs to be accepted. Subsequently, premiums are calculated [3]. On average, it takes at least 30 days for an application to be processed. However, nowadays, people are reluctant to buy services that are slow. Due to the underwriting process being lengthy and time-consuming, customers are more prone to switch to a competitor or prefer to avoid buying life

insurance policies. Lack of proper underwriting practices can consequently lead to customers being unsatisfied and a decrease in policy sales. The underwriting service quality is an essential element in determining the corporate reputation of life insurance businesses and helps in maintaining an advantageous position in a competitive market [4]. Thus, it is crucial improving the underwriting process to enhance customer acquisition and customer retention. Similarly, underwriting process and the medical procedures required by the insurance company to profile the risks of the applicants can be expensive [5]. Usually, all the costs to perform the medical examinations are initially borne by the firm. Underwriting costs are fully paid from the contract and can last 10–30 years. In case, where there is a policy lapse, the insurer incurs great losses [6]. Therefore, it is imperative to automate the underwriting process using analytical processes. Predicting the significant factors impacting the risk assessment process can help to streamline the procedures, making it more efficient and economical. A study by [7] shows that low underwriting capacities are a prominent operational problem among insurance companies surveyed in Bangladesh. Another threat to the life insurance businesses is that they can face adverse selection. Adverse selection refers to a situation where the insurers do not have all information on the applicant, and they end up giving life insurance policies to customers with a high-risk profile [8]. Insurance firms with competent underwriting teams stress on making the least possible losses. In other words, the insurers strive to avoid adverse selection as it can have powerful impacts on the life insurance business [9]. Adverse selection can be avoided by correctly classifying the risk levels of individual applications through predictive analytics, which is the goal of this research.

2.3.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to object detection and recognition of objects from 1985 to 2020 have been shown which discussed about different methods and algorithm.

Chapter 3. Proposed System

Proposed System

3.1 The Proposal

The proposal is to deploy a system which can identify various risk level of a particular applicant at the primary stage that may result in betterment of insurance industry.

It analyzes the past company data and customer information using Supervised Machine Learning approach. This system train the past data using the model that gives best accuracy and having less execution time in order to analyze the customer information and predicting their risk levels. Further this system accept/reject the applicant based the risk level predicted by the system

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by

this system:

- **Economic:** The proposed system is economic as there will not be any insurer required to collect the information of policy holder and calculating risk level. Risk can be predicted automatically by just providing new customer information to the system.
- **Man Power:** It does not require any person or their efforts to collect customer information and calculating risk manually.
- **24 x 7 Availability:** The proposed system will be available 24x7 for the customer. They just need to login onto their account and fill the insurance form anytime.
- **Statistical analysis:** The applicant data can be analyzed individually include height, weight, bmi, family history, insurance history etc. using statistical approach such as bar graph, pie chart, line graph etc.

3.3 Block Diagram

Figure 3-1: Block Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

For automating the analysis of risk factors of insurance policy holders using customer information, there is a need to apply predictive modelling approach. For this, the kind of framework used must be the one that is capable of predicting the risk levels easily and accurately based on the information provided. This can be done by train our model in the available past insurance data that further helps in predicting new customer risk. For this using predictive modeling approach, makes the system technically feasible.

The system once set up completely, works automatically without needing any person to operate it. The result (accept/reject based on risk levels), gets automatically saved in the database, without requiring any manual effort for saving it.

For making the system technically feasible, there is a requirement of selecting predictive model that gives more accurate result and has less execution time for better performance.

3.4.2 Economical

Our system is automatically predicting the risk of policy holders, there is a need of electricity supply.

This system is not expensive as it does not include any hardware component other than computer system and its peripherals.

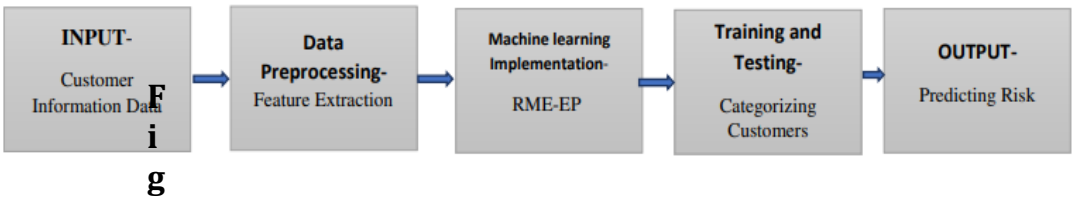
In this project we are using predictive modeling techniques that predicts the significant factors impacting the risk assessment process can help to streamline the procedures, making it more efficient and economical.

3.4.3 Operational

The main motto of our system is to reduce the manual efforts of collecting information of policy holders that impact risk assessment and analyzing those factors.

The system is able to do that accurately and efficiently making the system operationally feasible.

3.5 Design Representation



Fig

Figure 3-2: System Logic Architecture

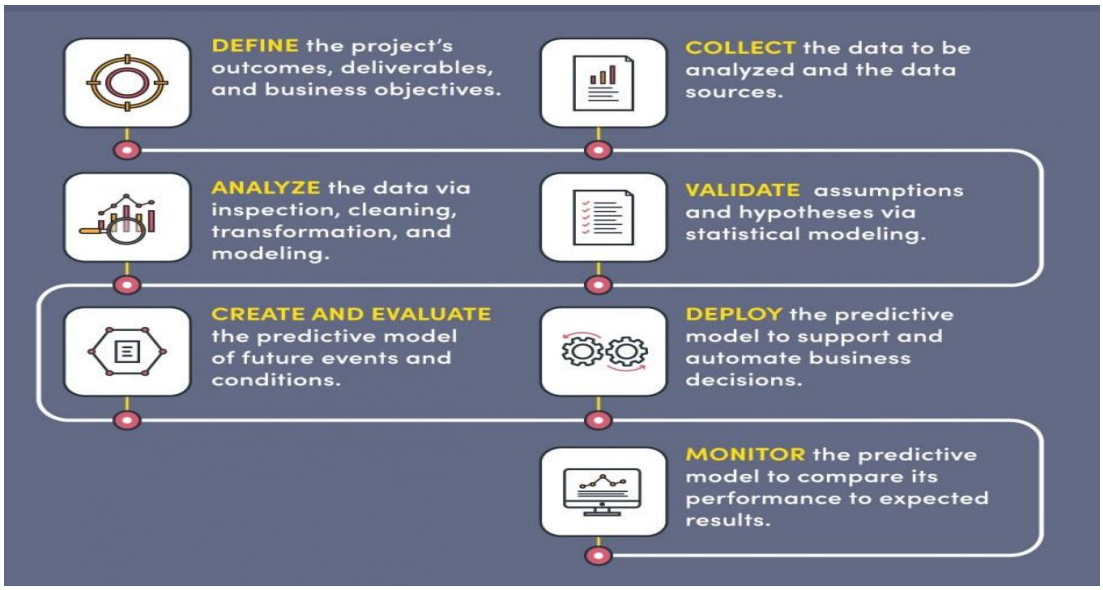


Figure 3-3: Complete System Architecture

Fig. 3-2 shows the system logic architecture of our project. We have collected the past customer dataset from kaggle. Further we have applied data preprocessing in order to extract relevant features that may be beneficial for risk assessment. Predictive modelling is applied for risk assessment which include KNN, Naive Bayes, SVM, Random Forest, Logistic Regression. Out of all these model we have taken Random Forest as it gives the best accuracy among all. Finally, we have deployed our model using Django Framework.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Id	Product_Inf	Product_Inf	Product_Inf	Product_Inf	Product_Inf	Product_Inf	Product_Inf	Ins_Age	Ht	Wt	BMI	Employmen	Employmen
2	1	1	D3	26	0.4871795	2	3	1	0.6119403	0.7818182	0.3389121	0.4722616	0.15	3
3	2	1	D3	10	0.0769231	2	1	1	0.641791	0.5818182	0.1485356	0.323008	0.028	12
4	3	1	A2	26	0.0769231	2	3	1	0.6268657	0.7272727	0.3117155	0.484984	0	1
5	4	1	D3	26	0.1446667	2	3	1	0.5820896	0.7090909	0.3200837	0.5191032	0.143	9
6	5	1	A1	26	0.0769231	2	3	1	0.0597015	0.6	0.1317992	0.2722877	0	1
7	6	1	E1	26	0.0769231	2	3	1	0.0298507	0.7454545	0.2887029	0.4287804	0.03	9
8	7	1	D4	10	0.4871795	2	3	1	0.1641791	0.6727273	0.2050209	0.3524377	0.042	9
9	8	1	D2	26	0.2307692	2	3	1	0.4179104	0.6545455	0.2343096	0.4240456	0.027	9
10	9	1	A1	26	0.1517087	2	1	1	0.5223881	0.6545455	0.2677824	0.4869621	0.21	9
11	10	1	D2	26	0.2307692	3	1	1	0.5074627	0.8363636	0.2991632	0.3648867	0.325	15
12	11	1	A8	10	0.1661938	2	3	1	0.3731343	0.5818182	0.1736402	0.3765867	0.11	1
13	12	1	A1	26	0.0769231	2	3	1	0.2985075	0.6727273	0.2468619	0.4287182	0.085	9
14	13	1	D3	26	0.2307692	2	3	1	0.5671642	0.8181818	0.2991632	0.3797544	0.075	9
15	14	1	D2	26	0.0769231	2	3	1	0.6119403	0.7818182	0.4037657	0.5716115	0.12	12
16	15	1	D3	26	0.2307692	2	3	1	0.5223881	0.6181818	0.1841004	0.3626431	0.165	9
17	16	1	E1	21	0.0769231	2	3	1	0.5522388	0.6	0.2845188	0.5877958	0.025	1
18	17	1	D3	26	0.1282051	2	3	1	0.5373134	0.6909091	0.3096234	0.5216685	0.05	9
19	18	1	D4	26	0.2307692	2	3	1	0.2985075	0.6909091	0.2719665	0.4550501	0.09	3
20	19	1	A2	26	0.1025641	2	3	1	0.5671642	0.6181818	0.1631799	0.320784	0.075	9
21	20	2	D1	26	0.4871795	2	3	1	0.2238806	0.7818182	0.3619247	0.5075148	0.1	9
22	21	1	A3	26	1	2	3	1	0.2985075	0.6909091	0.2259414	0.3736277	0.14	9
23	22	1	D4	26	0.4871795	2	3	1	0.3283582	0.6363636	0.1422594	0.2646482	0.16	3
24	23	1	A7	26	0	2	3	1	0.6268657	0.6727273	0.3305439	0.581279	0.075	9
25	24	2	D4	26	0.4871795	2	3	1	0.2089552	0.7454545	0.2468619	0.3609687	0.1	14
26	25	1	D3	26	0.3846154	2	3	1	0.2686567	0.6363636	0.2280335	0.4309492	0.0378	9
27	26	1	D3	26	0.0769231	2	3	1	0.3880597	0.7818182	0.3096234	0.4273938	0.08	9
28	27	1	D4	26	0.4871795	2	3	1	0.2238806	0.6	0.1380753	0.2852538	0.055	9
29	28	1	D4	26	0.2564103	2	3	1	0.0597015	0.6545455	0.2154812	0.3886551	0.025	9

Table 1: Datasets

Table 1 shows the dataset that we have taken from kaggle. In this dataset we have hundreds of variables describing attributes of insurance policy applicants.

Following are its data fields:

- Id: A unique identifier associated with an application.
- Product_Info_1-7: A set of normalized variables relating to the product applied for
- Ins_Age: Normalized age of applicant
- Ht.: Normalized height of applicant
- Wt.: Normalized weight of applicant
- BMI: Normalized BMI of applicant
- Employment_Info_1-6: A set of normalized variables relating to the employment history of the applicant.
- InsuredInfo_1-6: A set of normalized variables providing information about the applicant.
- Insurance_History_1-9: A set of normalized variables relating to the insurance history of the applicant.
- Family_Hist_1-5: A set of normalized variables relating to the family history of the applicant.

- Medical_History_1-41: A set of normalized variables relating to the medical history of the applicant.
- Medical_Keyword_1-48: A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
- Response: This is the target variable, an ordinal variable relating to the final decision associated with an application

3.5.1 Use Case Diagram

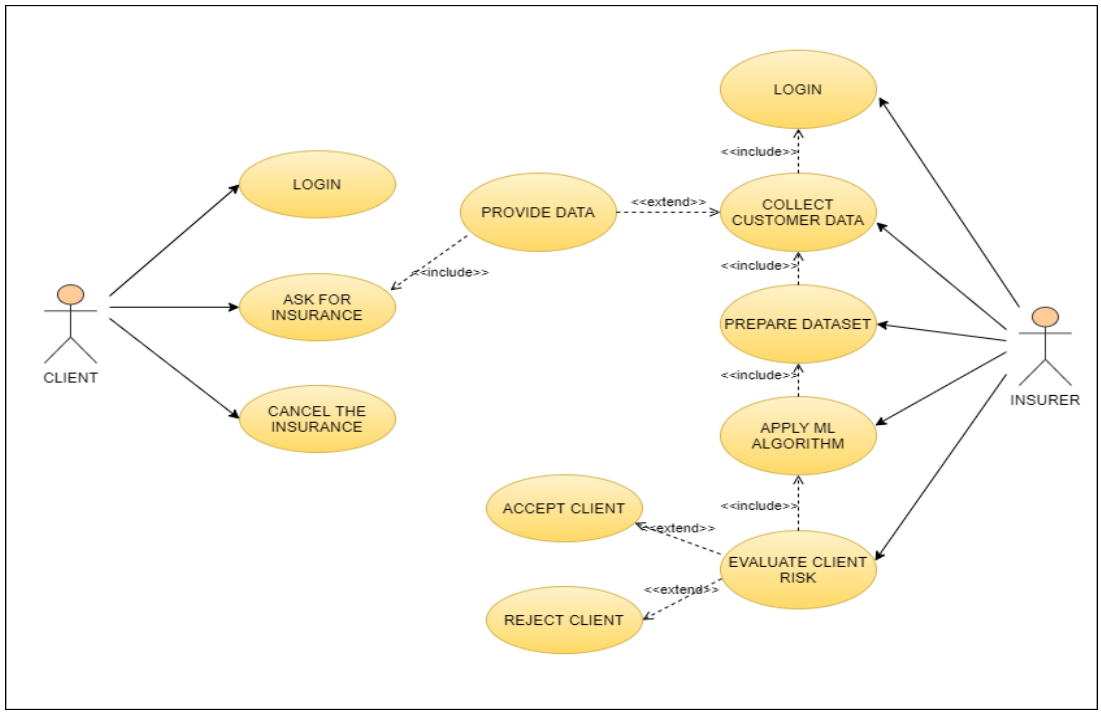


Figure 3-4 Use Case Diagram

3.5.2 ER Diagram

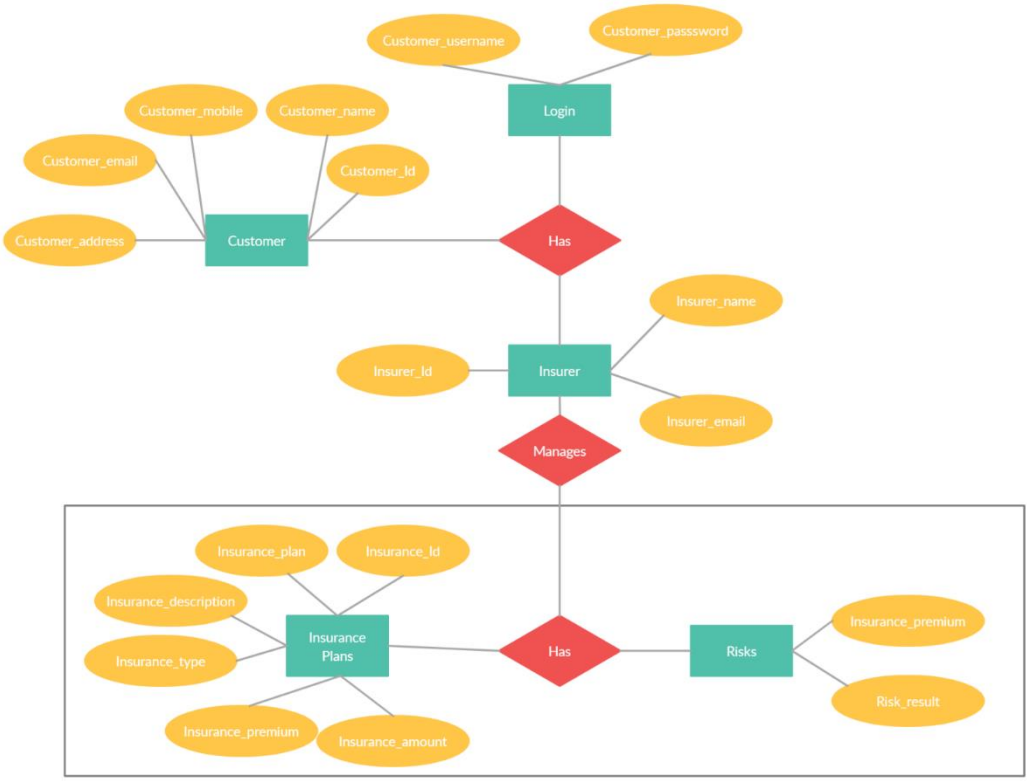


Figure 3-5 ER Diagram

3.5.3 Class Diagram

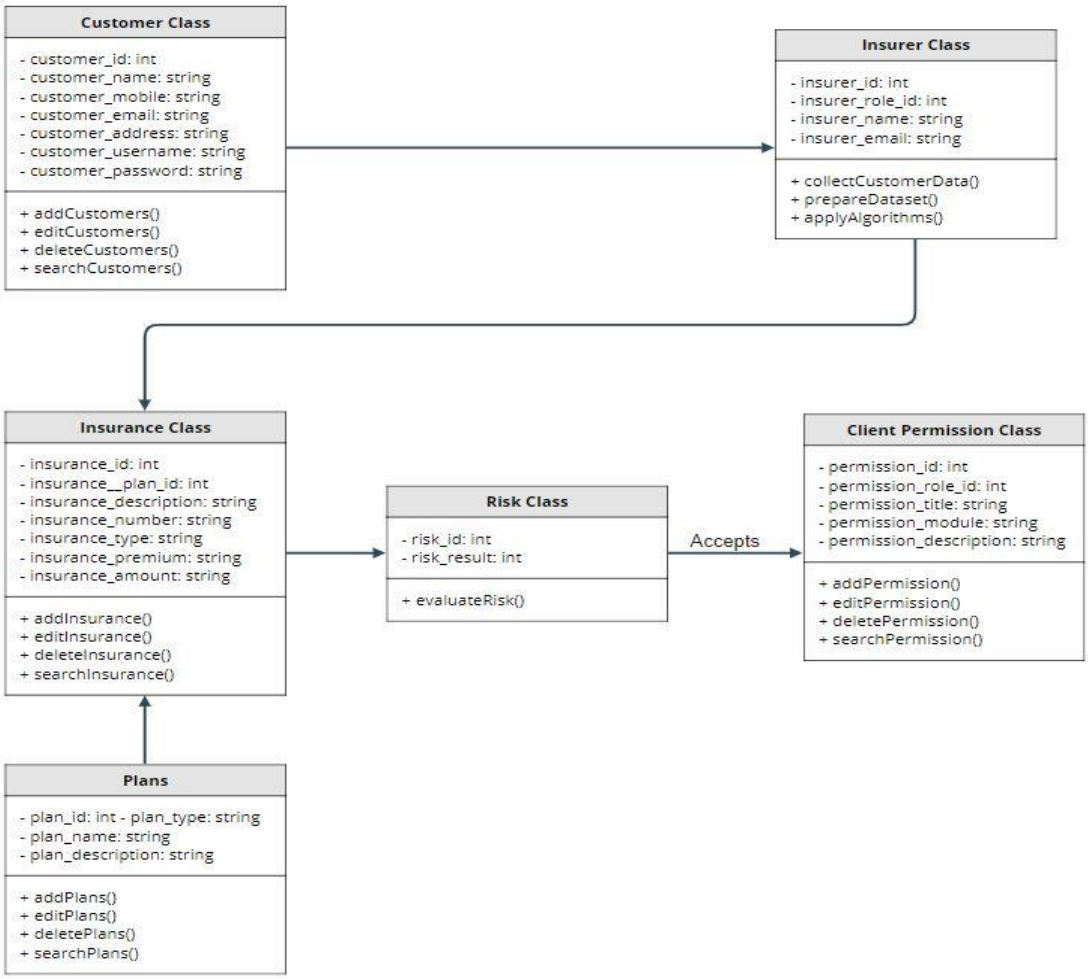


Figure 3-6 Class Diagram

3.5.4 Object Diagram

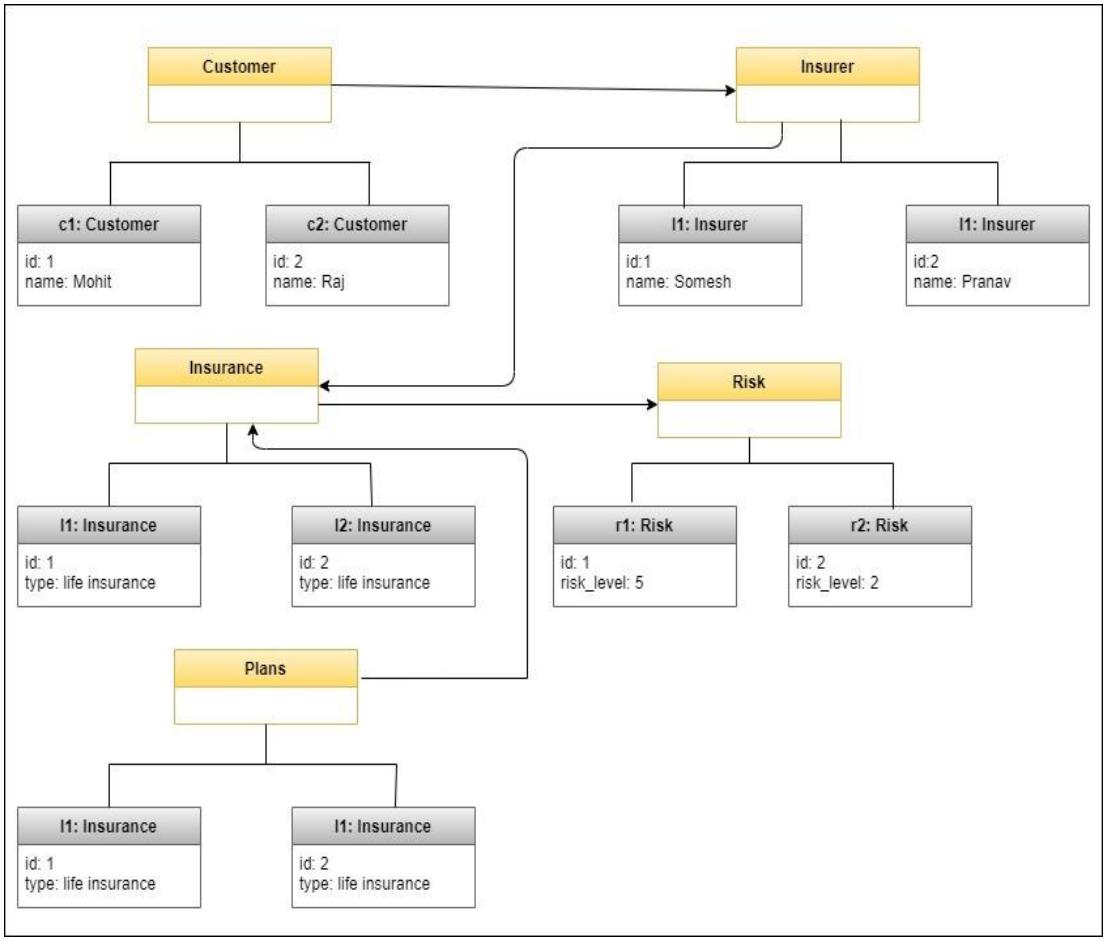


Figure 3-7 Object Diagram

3.5.5 Data Flow Diagram

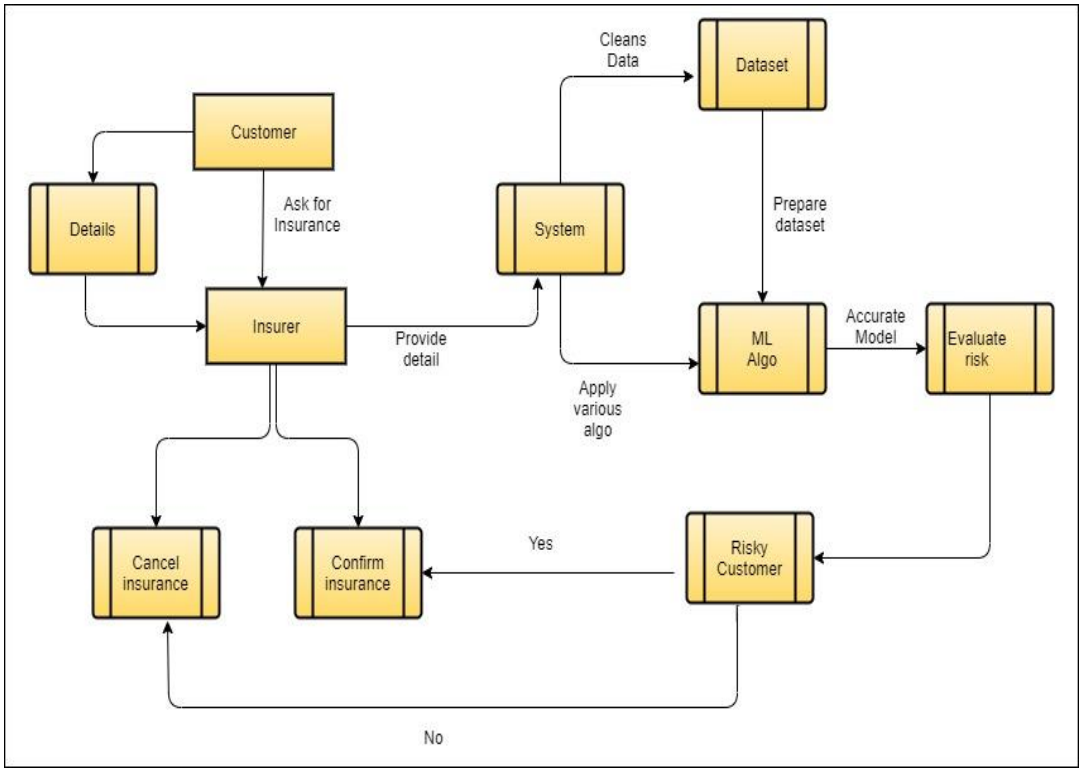


Figure 3-8 Data Flow Diagram

3.5.6 Sequence Diagram

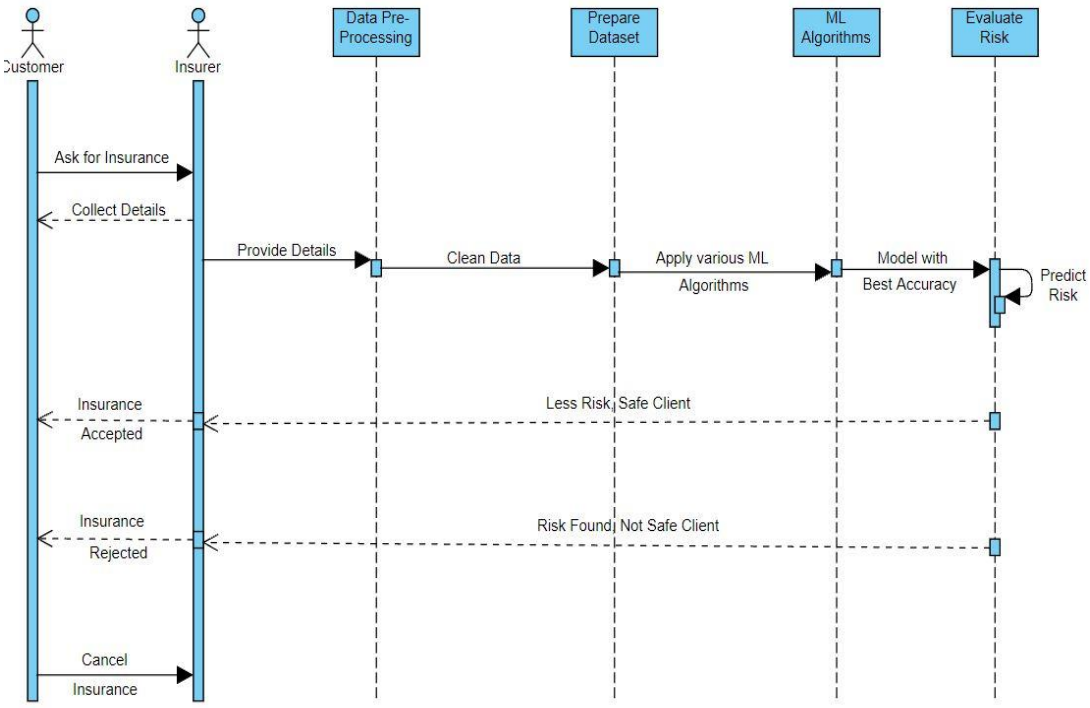


Figure 3-9 Sequence Diagram

3.5.7 Activity Diagram

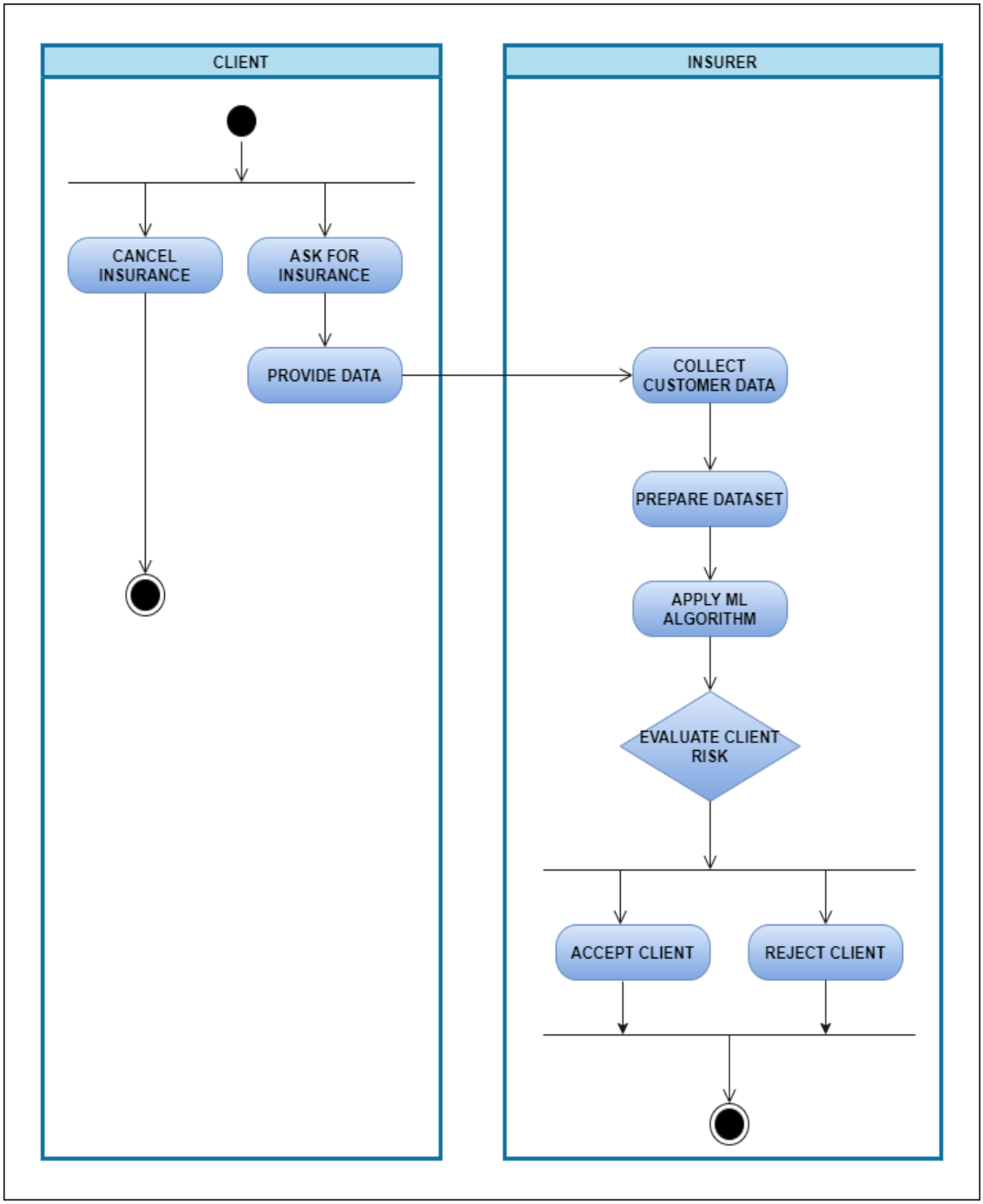


Figure 3-10 Activity Diagram

3.5.8 Deployment Diagram

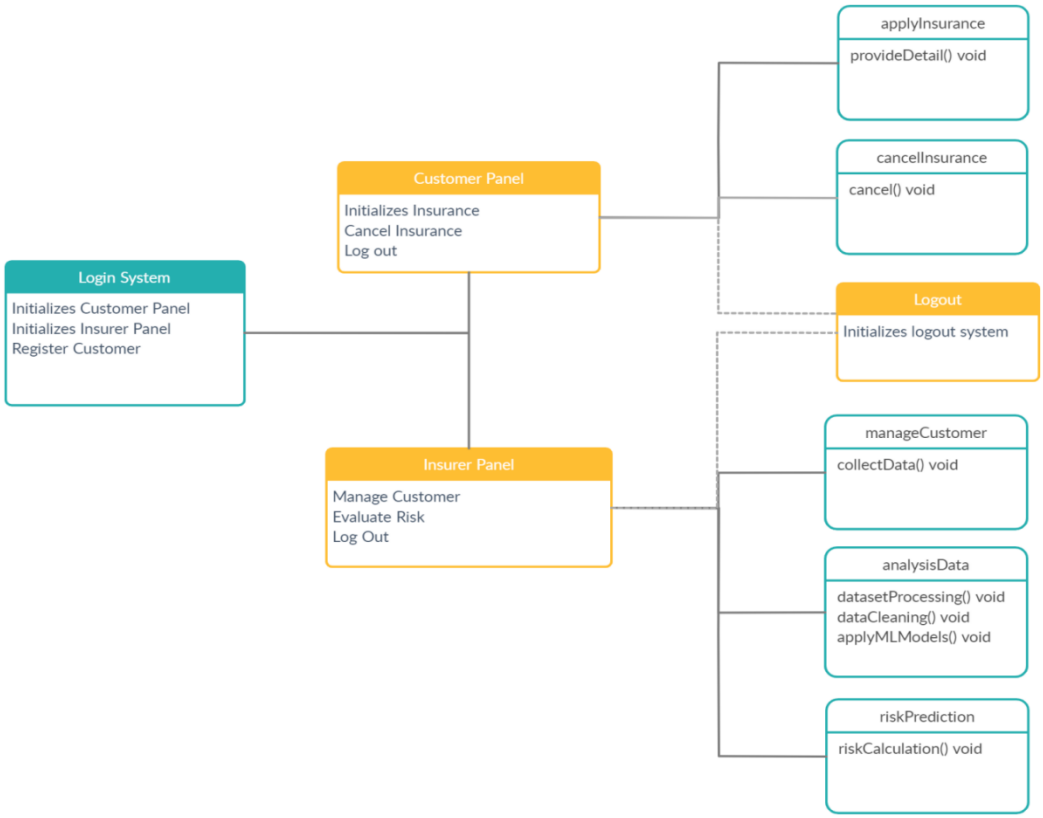


Figure 3-11 Deployment Diagram

3.5.2 Database Structure

The name of the database created is “db_detect” and there is one table in the database named “logs” for storing the records.

The “Logs” table has the following structure :

Name	Data Type	Description
Datetime	Timestamp	Shows the complete date and time when the person/vehicle enters and is identified
Type	Varchar2	Displays the type of object for example Person, Car, Dog.
CIF	Number	Count per frame.It tells the number of objects in frame.

Table 2 : Database Structure

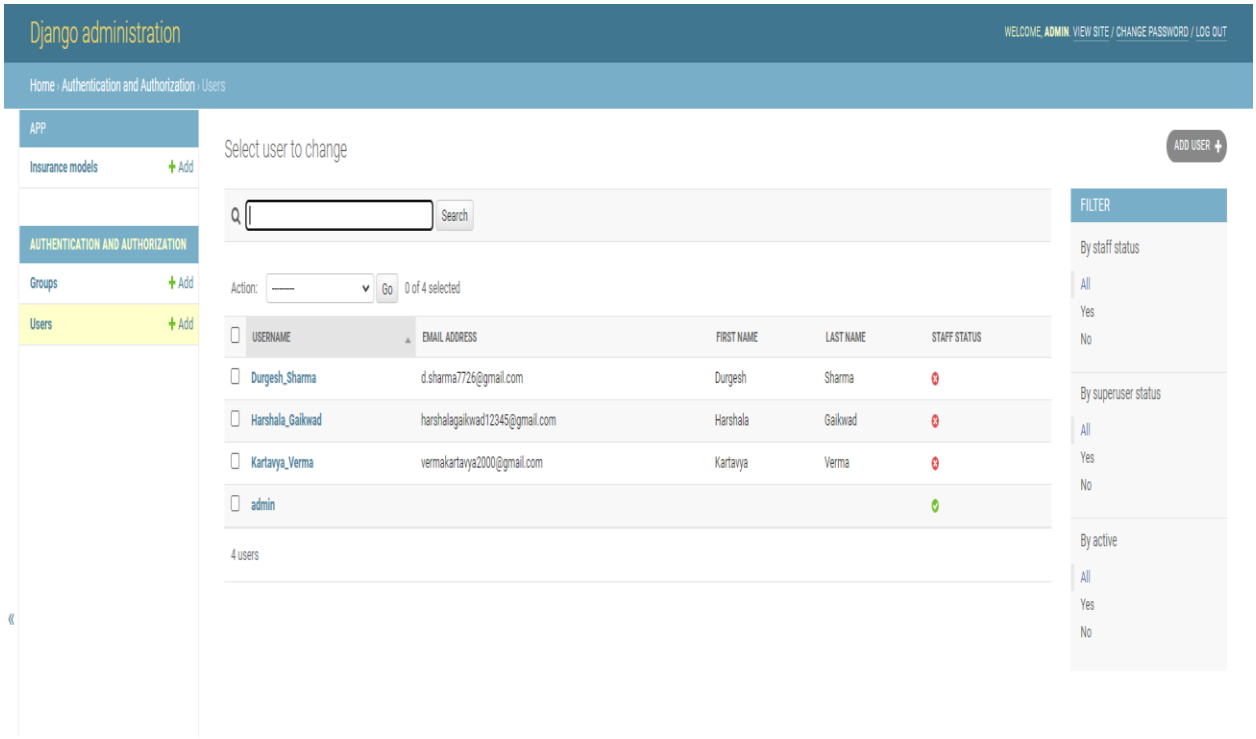


Table 3 : Database Structure of Users

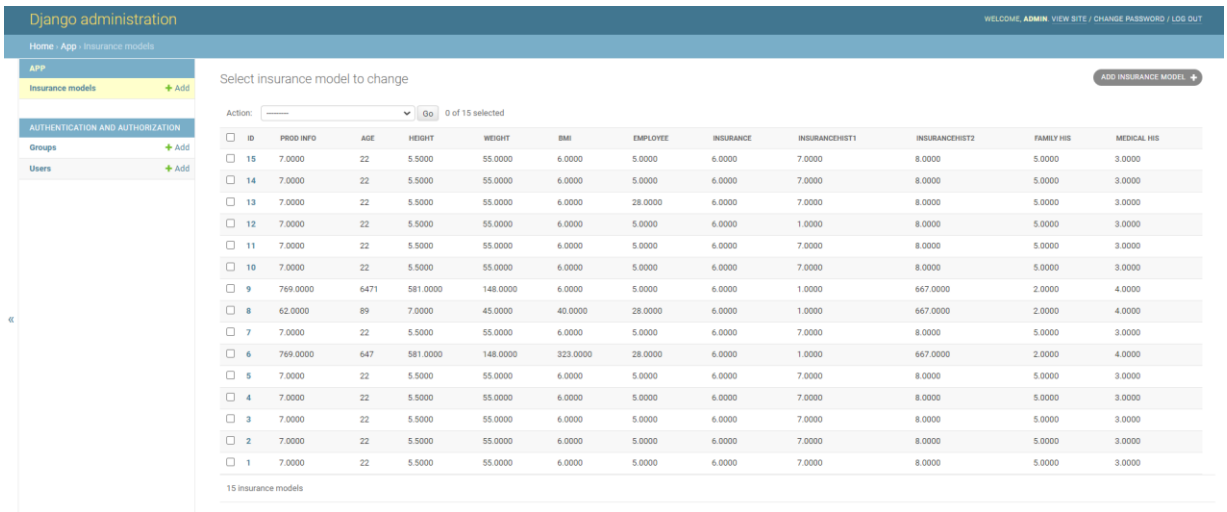


Table 4 : Database Structure of Customer Information

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

3.6.1 Hardware

- Processor – 1.9 gigahertz(GHz) x86 or x64 bit dual core processor with SSE2 instruction set
- Display – Super VGA with a resolution of 1024 x 768
- Memory – 4 GB RAM
- Network Requirements -
    1. Bandwidth greater than 50 Kbps (400 kbps)
    2. Latency under 150 ms

3.6.2 Software

- Python IDLE 3.7 and its supported libraries
- Pycharm and Anaconda
- Library installation:
    - Numpy
    - Pandas
    - Sklearn
    - Matplotlib
    - Seaborn
    - Django
- Self-Hosted Technical Requirement-
    1. OS – Windows 10
    2. Browser – Chrome v77+

Chapter 4. Implementation

Implementation

The main foundational block of insurance industry is to estimate the future events and measure the associated risk of these events, hence it is needless to say that predictive analytics is used widely to determine the risk factor associated with the applicant. Analyzing thousands of applicant data manually may take large amount of time due to these reason insurance industries are considered as slow-moving industry.

With the advent of advanced data analytics technologies, we can overcome this problem by introducing the system that uses predictive modelling to determine the risk associated with the customer based on their behavior which are indicated by different attributes such as height, age, weight, BMI (Body Mass Index), family history etc. The data collected from an individual helps insurance companies in enhancing pricing and risk selection.

Additionally, it helps in making better decisions, understanding customer needs and be fair to the customers. Acquiring a comprehensive understanding of customer behaviors and habits from historical data helps insurers to anticipate future behaviors and provide the right insurance product and policy premium.

4.1 Technique Used

4.1.1 Supervised Machine Learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

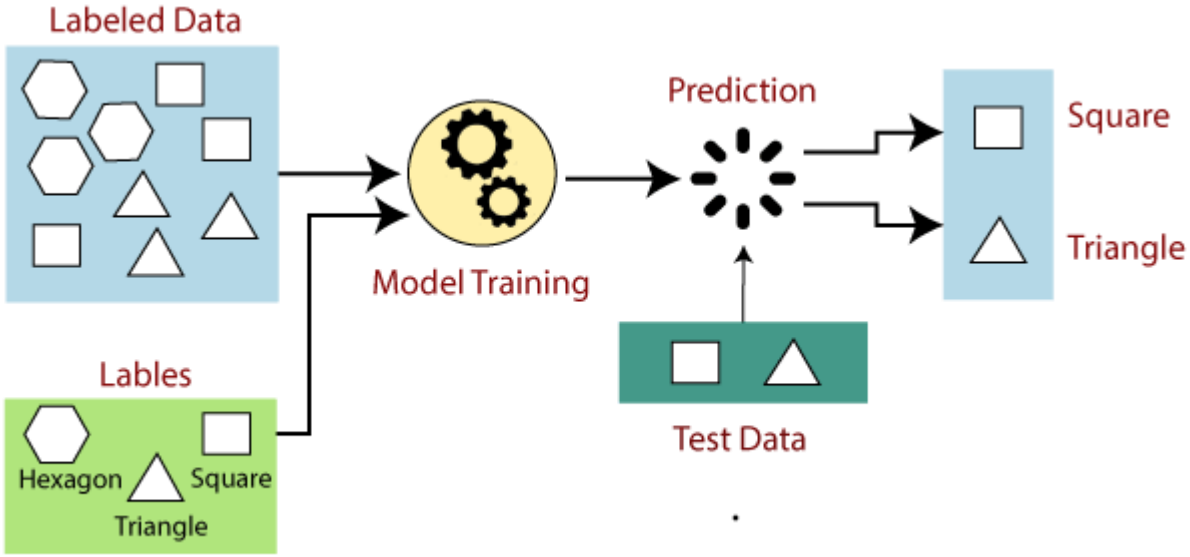


Figure 4-1: Supervised Machine Learning

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

In the real world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

In our project we are using supervised machine learning techniques. The dataset collected from kaggle website is a labelled data that contains risk levels (1-8) as a response variable. We train our model using this dataset and try to predict the new customer's risk level.

4.1.2 Predictive Modelling Techniques:

Predictive Modelling is the process of uncovering relationship within the data by using a mathematical model for predicting some desired outcome. It uses historical data to make predictions about unseen data.

Figure 4-2: Steps for predictive modelling

Fig. 4-2 depicts the cycle of the predictive modelling. The goal of our project is to predict the risk level which is between 1 to 8. This may be further classify as accepted (1) or rejected (0). So, binary classification models like logistic regression, random forest, K-Nearest Neighbors(KNN), Naïve Bayes and Support Vector Machine(SVM) are used for predictive modelling.

To predict the risk level of different applicants we have use following predictive models:

1. **RANDOM FOREST**

Random forest is a Supervised Machine Learning Algorithm that

is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Following is the result when K-fold cross validation is applied into the dataset using Random Forest:

```
[ ] PLI_RF_7 = RandomForestClassifier(n_estimators = 200, criterion = 'gini', max_features = None)
    scores = cross_val_score(PLI_RF_7, X_train, Y_train, cv = 10, scoring = 'accuracy')
    print(scores)
    print(scores.mean())

    [0.82757072 0.82824427 0.83004041 0.831163   0.82801976 0.82910397
     0.82843027 0.84527285 0.82506176 0.82169324]
    0.8294600260909961
```

Accuracy: 0.8294600260909961

2. LOGISTIC REGRESSION

Logistic regression is a Supervised Learning technique. It predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Following is the result when K-fold cross validation is applied into the dataset using Logistic Regression:

```
[ ] PLI_LogR = LogisticRegression(penalty = 'l2', C=1.0,solver='lbfgs', max_iter=1000)
    scores = cross_val_score(PLI_LogR, X_train, Y_train, cv = 10, scoring = 'accuracy')
    print(scores)
    print(scores.mean())

    [0.81454872 0.80669062 0.80489448 0.81365065 0.81185451 0.81473164
     0.81495621 0.8198967  0.80777004 0.80282955]
    0.8111823121721041
```

Accuracy: 0.8111823121721041

3. K NEAREST NEIGHBORS(KNN)

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

Following is the result when K-fold cross validation is applied into the dataset using KNN:

```
[ ]
    KNN_6 = KNeighborsClassifier(n_neighbors = 100, algorithm = 'brute')
    scores = cross_val_score(KNN_6, X_train, Y_train, cv = 10, scoring = 'accuracy')
    print(scores)
    print(scores.mean())
```

```
[0.69039066 0.68320611 0.69330938 0.68343062 0.68500225 0.68582978
 0.6752751  0.68874916 0.67841904 0.6842578 ]
0.684786989916436
```

Accuracy: 0.684786989916436

4. NAIVE BAYES

Naive Bayes is a classification algorithm that is suitable for binary and multiclass classification. It is a supervised classification technique used to classify future objects by assigning class labels to instances/records using conditional probability.

Following is the result when K-fold cross validation is applied into the dataset using Naive Bayes:

Navie Bayes

```
[ ] PLI_GNB = GaussianNB()
    scores = cross_val_score(PLI_GNB, X_train, Y_train, cv = 10, scoring = 'accuracy')
    print(scores)
    print(scores.mean())
```

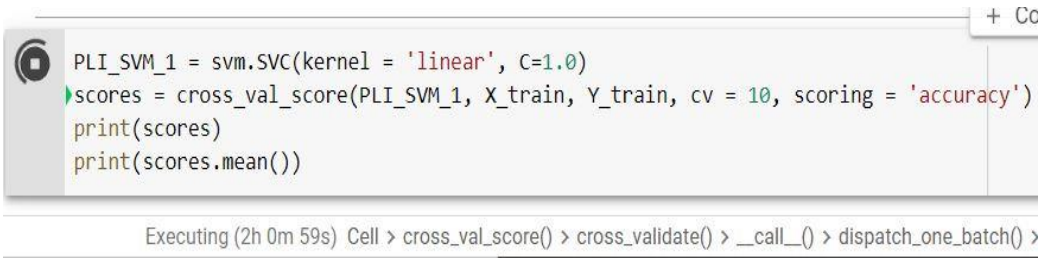
```
[0.64369106 0.655815   0.6506511  0.64795689 0.65671307 0.65147092
 0.64720413 0.65416573 0.64787784 0.64203907]
0.649758481313234
```

Accuracy: 0.649758481313234

5. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. In SVM, we plot each data item as a point in n-dimensional space (n is number of features) with the value of each feature being the value of a particular coordinate. Then we perform classification by finding hyper-plane that differentiates the two classes.

Following is the result when K-fold cross validation is applied into the dataset using SVM:



This model takes a lot of computational time (approximately more than 2 hours) as compared to others.

4.2 Tools Used

4.2.1 Numpy

NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices. Arrays are very frequently used in data science, where speed and resources are very important.

In Python we have lists that serve the purpose of arrays, but they are slow to process.NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

4.2.2 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

4.2.3 Sklearn

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.

It has various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.2.4 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.SciPy makes use of Matplotlib.

4.2.5 Django

It is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself.[9] Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

4.3 Language Used

Python language is used in the system due to the following Characteristics:

Simple:

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source:

Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented:

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable

pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

Extensive Libraries:

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python.

4.4 Screenshots

The Following are the screenshots of the result of the project :

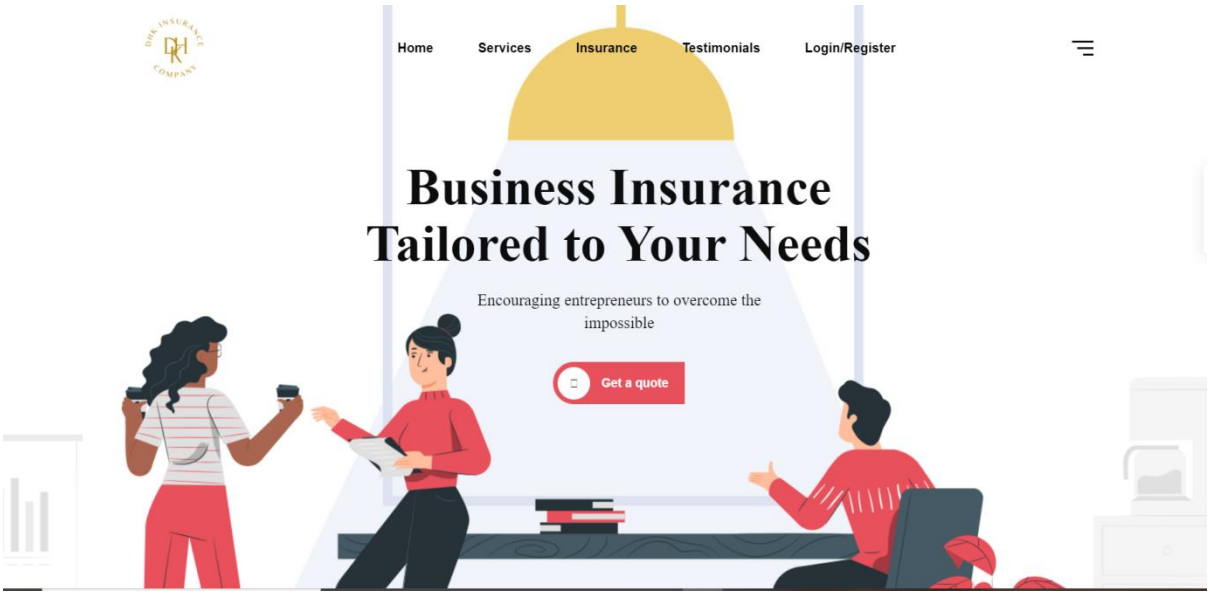


Figure 4-3 : Screenshot 1

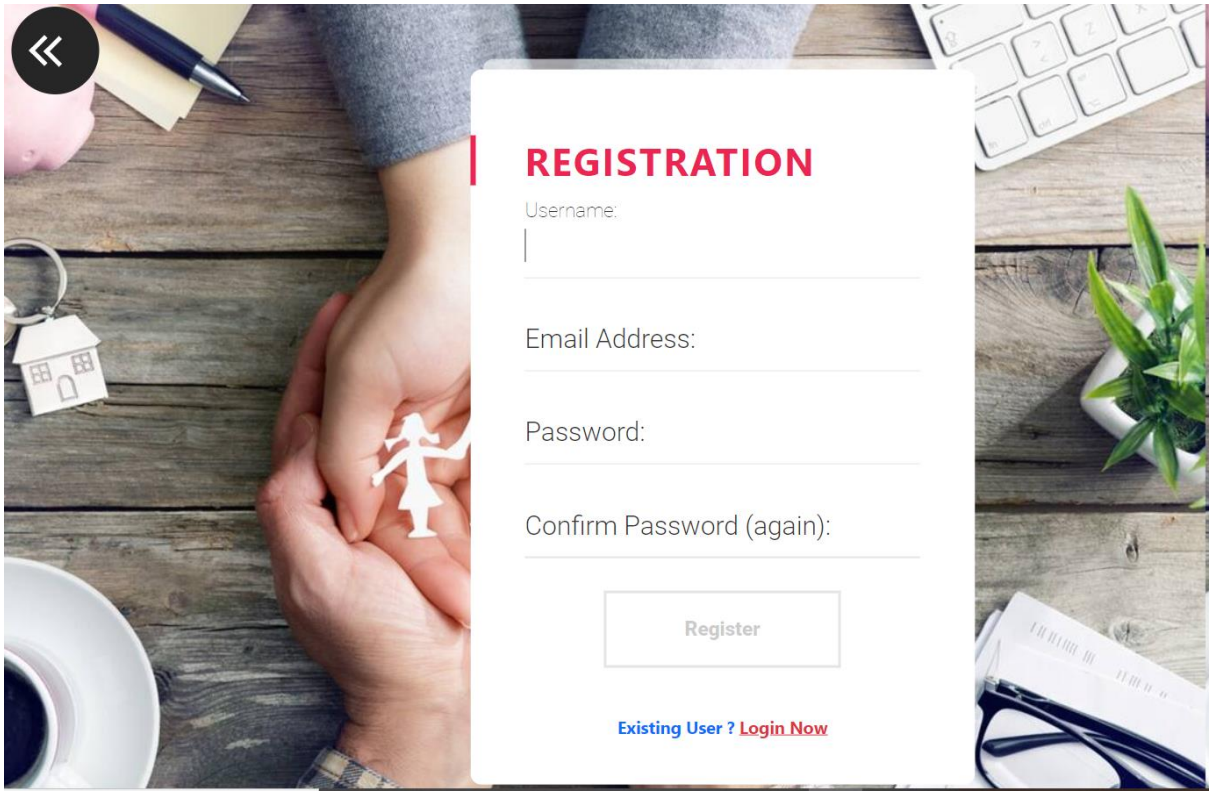


Figure 4-4 : Screenshot 2

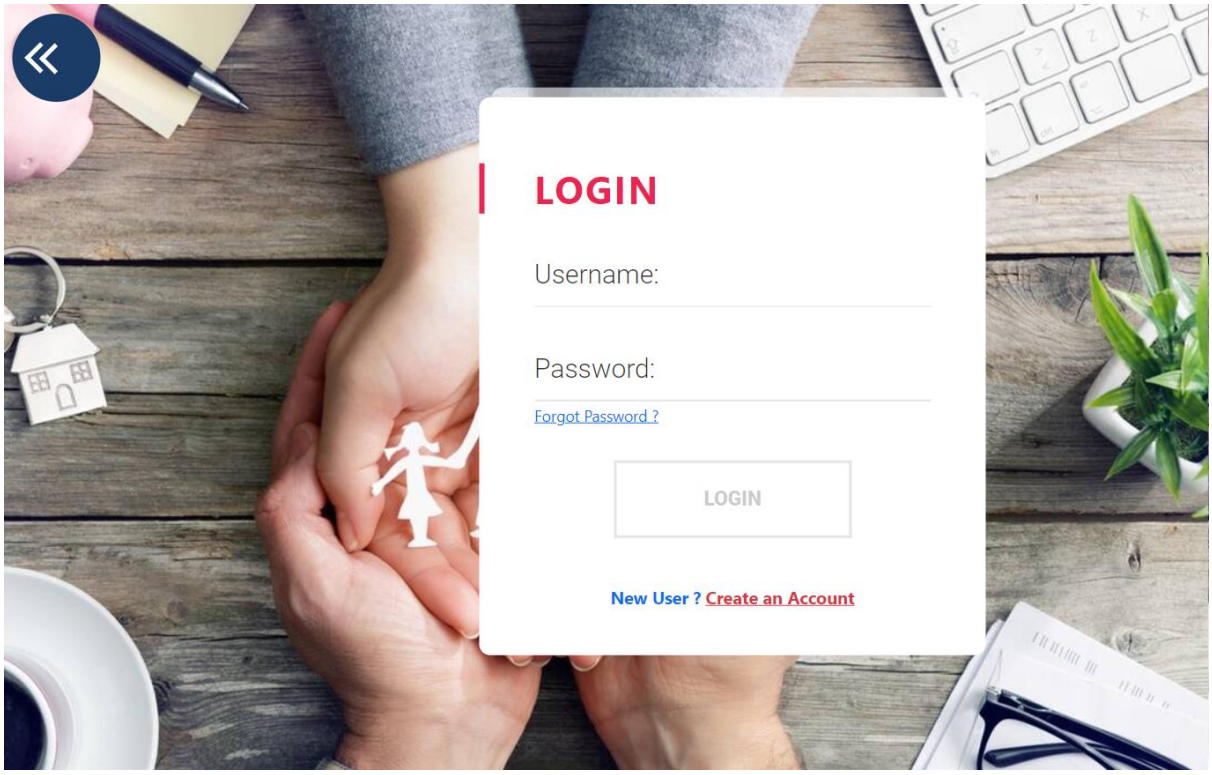


Figure 4-5 : Screenshot 3

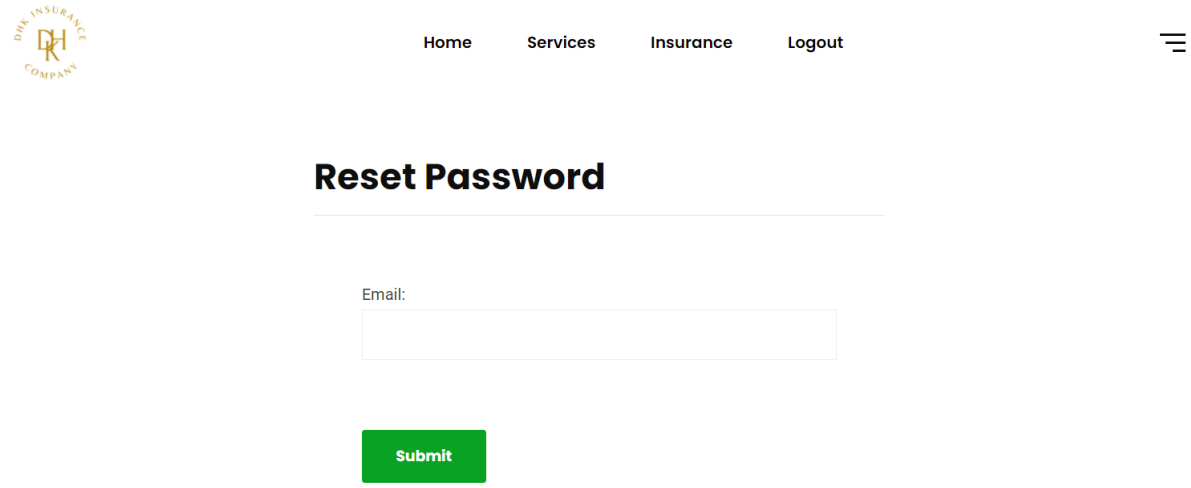


Figure 4-6 : Screenshot 4

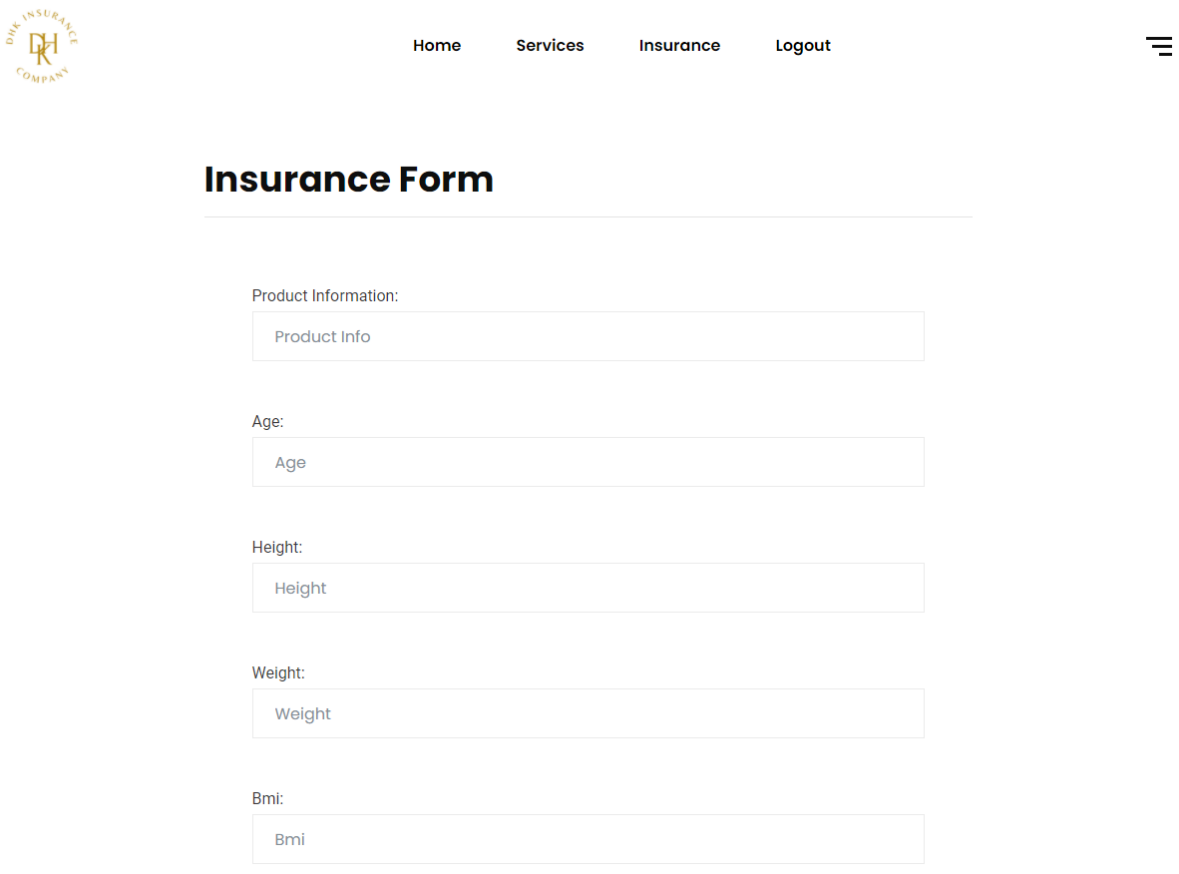


Figure 4-7 : Screenshot 5

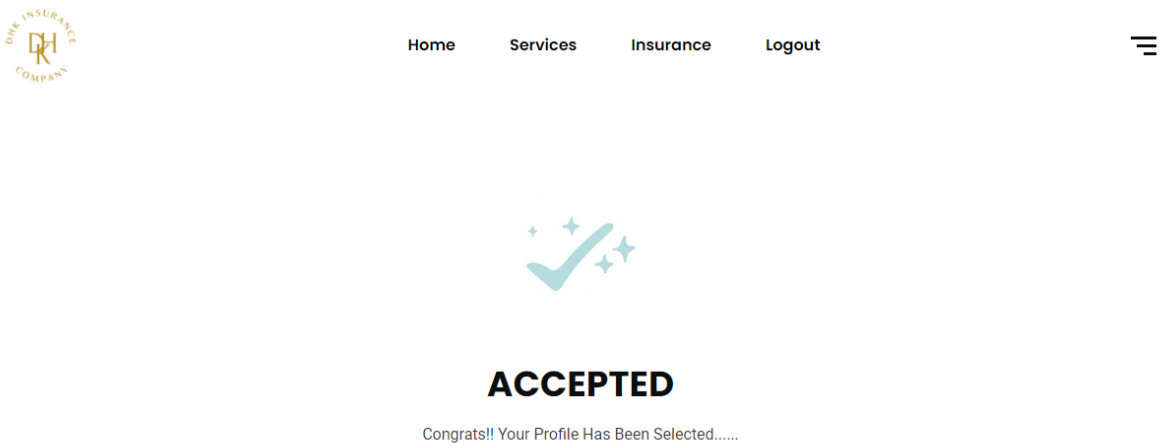


Figure 4-8 : Screenshot 6

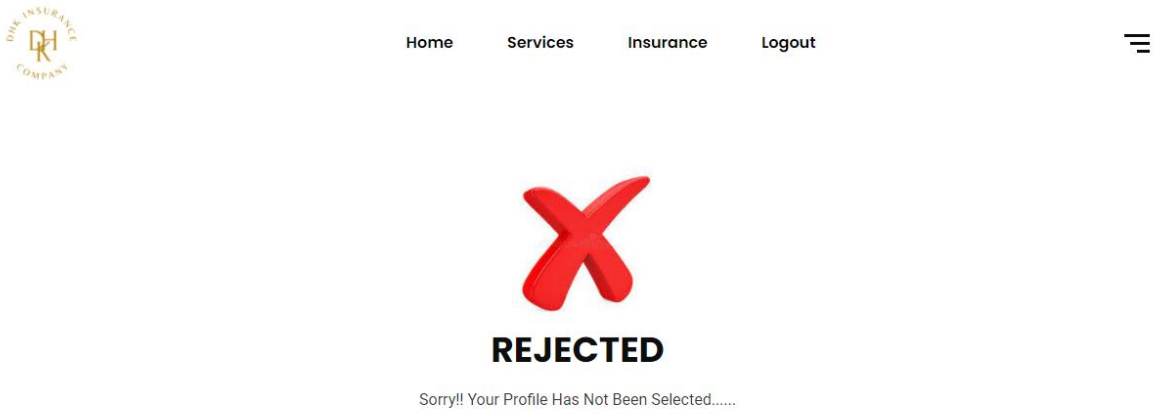


Figure 4-9 : Screenshot 7

4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process. .

4.5.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

There's some tests that we run without needing trained parameters. These tests are:

We checked the shape of our model output and ensure it aligns with the labels in our dataset.

We checked the output ranges and ensure it aligns with our expectations (eg. the output of a classification model should be a distribution with class probabilities that sum to 1)

Make sured a single gradient step on a batch of data yields a decrease in our model loss

Checked for label leakage between our training and validation datasets.

4.5.2 Test Case and Analysis

TEST CASE: 1

Test Case ID	TC001
Test Case Summary	It will check the input values from the user and according to the values it will accept or reject the insurance policy as per the machine learning algorithm
Test Procedure	1.User enter the values in the insurance form. 2. Machine Learning Model gives the predicted result.
Expected Result	For each user,if the insurance policy is suitable according to the parameters, the result will be accepted.
Actual Result	Accepted
Status	Pass

Table 5 : Test Case 1

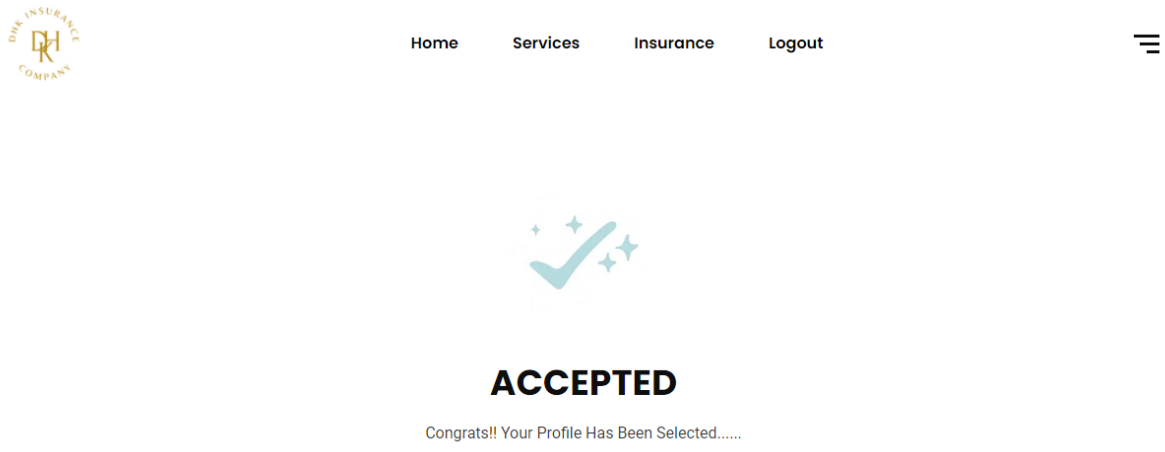


Figure 4-10 : Test Case 1 Result

TEST CASE: 2

Test Case ID	TC002
Test Case Summary	It will check the input values from the user and according to the values it will accept or reject the insurance policy as per the machine learning algorithm.
Test Procedure	1.User enter the values in the insurance form. 2. Machine Learning Model gives the predicted result.
Expected Result	For each user,if the insurance policy is not suitable according to the parameters, the result will be rejected.
Actual Result	Rejected
Status	Pass

Table 6: Test Case 2

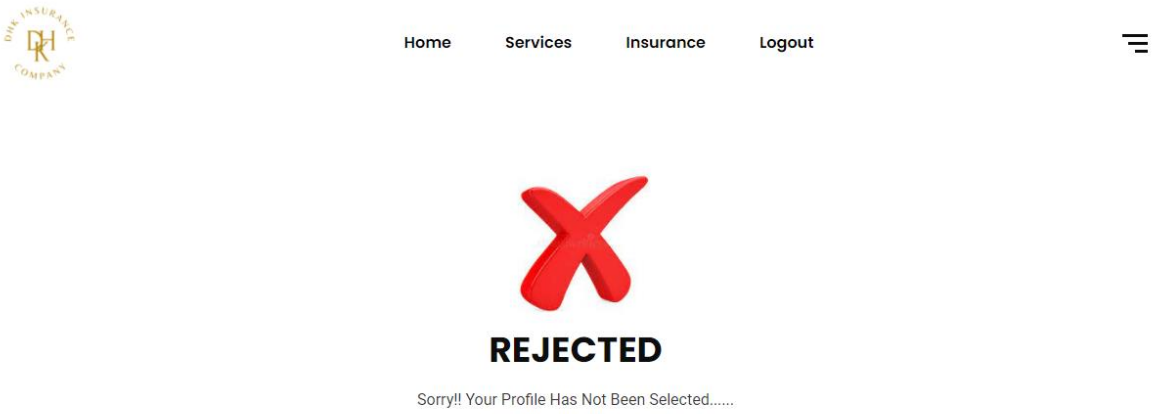


Figure 4-11 : Test Case 2 Result

Chapter 5. Conclusion

Conclusion

5.1 Conclusion

Data analytics is now the trend that is gaining significance among companies worldwide. In the life insurance domain, predictive modeling using machine learning algorithm scan provide the notable difference in the way which business is done as compared to the traditional methods. It has tremendous benefits in determining how much the premium should be charged to the individual based upon his/her behaviors and information provided by them. Insurance Company can accurately identify the risk level based upon a specific individual's attributes.

Previously, risk assessment for life underwriting was conducted using complex actuarial formulas and usually was a very lengthy process. Now, with data analytical solutions, the work can be done faster and with better results. Therefore, it would enhance the business by allowing faster service to customer, thereby increasing satisfaction and loyalty.

This will not only help the individuals in getting charged the right amount of premium for their insurance but will also help in forging better relationships and a level of trust between the insurance company and the insured. Based on these predictions, the health insurance providers can then evaluate the following decisions and make better judgement calls:

- Which individuals deserve which kind of insurance plan
- How much the premium should be charged based on an individual's behaviors?

- Based upon an individual's behavior, predicting their premium helps in better risk management.
- It helps forge trust between the customer and the insurance company.

  Thus, it is important for a insurance company to collect and analyze the data such as a person's age, BMI, height, weight, family and past insurance data to accurately predict the risk and charge accurate premiums to cover that risk.

5.2 Limitations of the Work

- The working of this project would be a little slow because framework like Django & deep learning need high-processing hardware and GPU(graphical processing unit) systems but we are using CPU only.
- The models that we are using are pre-trained models. So, if we want to train our own model, it takes a lot of time and processing.

- In the system, checking of each value for all different users needs improvement.

5.3 Suggestion and Recommendations for Future Work

- The Model would be trained for checking more number of customers.
- Insurance advice system will also be integrated in this system which tells users to select the appropriate insurance policy.
- Price optimization , personalized marketing and fraud detection features can be added in future.

Bibliography

[1] Mishr K (2016) Fundamentals of life insurance theories and applications. In: 2nd ed, Delhi: PHI Learning Pvt Ltd.

[2] Wuppermann A (2016) Private information in life insurance, annuity and health insurance markets. Scand J Econ 119:1–45.

[3] Prince A (2016) Tantamount to fraud? Exploring non-disclosure of genetic information in life insurance applications as grounds for policy rescission. Health Matrix 26:255–307.

[4] Chen TJ (2016) Corporate reputation and financial performance of life insurers. Geneva Papers Risk Insur Issues Pract 41:378–397.

[5] Huang Y, Kamiya S, Schmit J (2016) A model of underwriting and post-loss Test without commitment in competitive insurance market. SSRN Electron J.

[6] Carson J, Ellis CM, Hoyt RE, Ostaszewski K (2017) Sunk costs and screening: two-part tariffs in life insurance. SSRN Electron J 1–26.

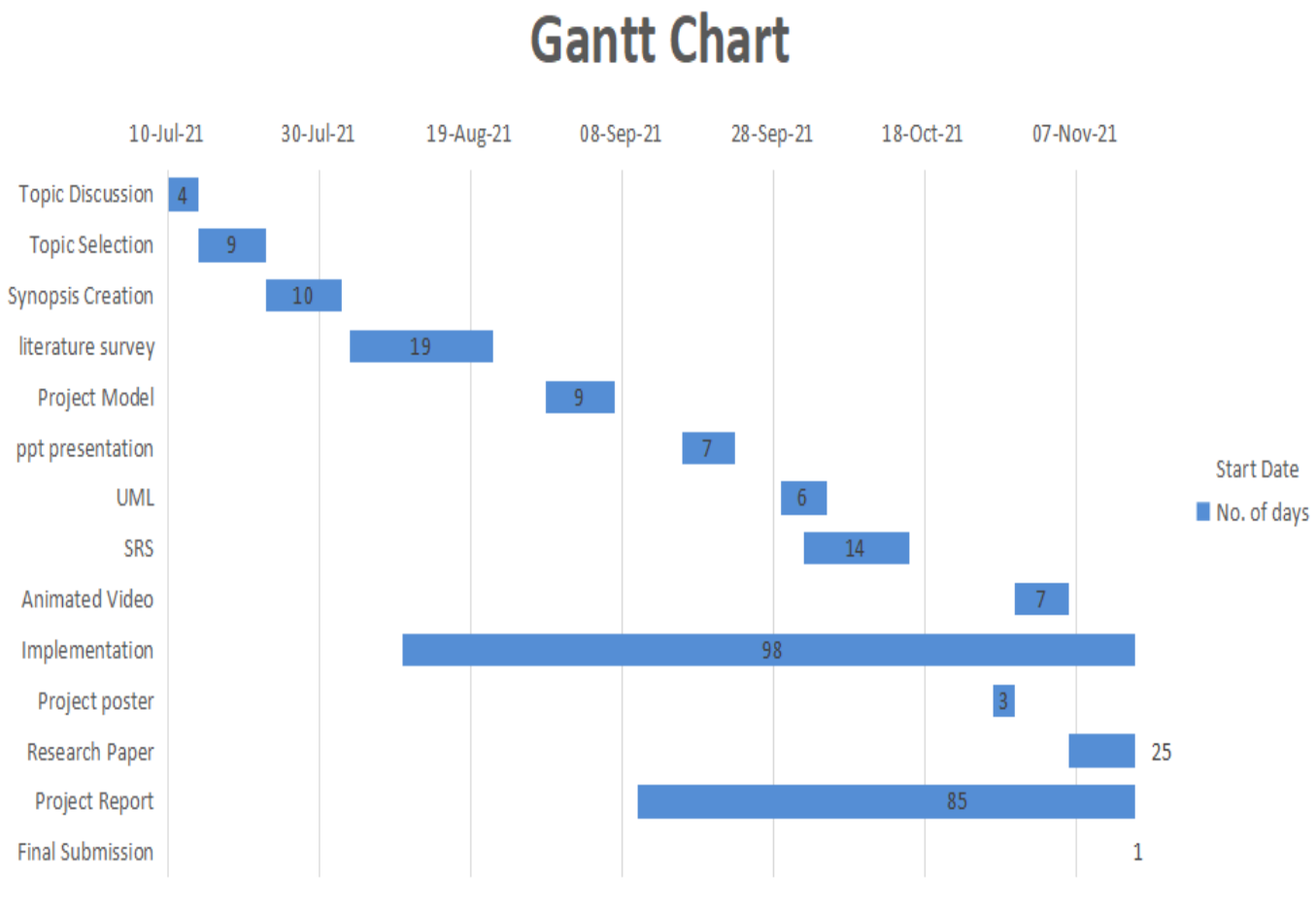
[7] Mamun DMZ, Ali K, Bhuiyan P, Khan S, Hossain S, Ibrahim M, Huda K (2016) Problems and prospects of insurance business in Bangladesh from the companies' perspective. Insur J Bangladesh Insurance Acad 62:5–164.

[8] Harri T, Yelowitz A (2014) Is there adverse selection in the life insurance market? Evidence from a representative sample of purchasers. Econ Lett 124:520–522.

[9] Hedengren D, Stratmann T (2016) Is there adverse selection in life insurance markets? Econ Inq 54:450–463.

Project Plan

Gantt Chart



Guide Interaction Sheet

Date	Discussion	Action Plan
4/08/2021	Discussed about the title of the Project	Predicting Risk for New Insurance using Customer Information as the title.
10/08/2021	Discussion on the technology to be used for object detection in real-time	Django , ML Model and other tools were finalized
14/08/2021	Discussion of the creation of synopsis of the project	Gathering of information for synopsis creation
17/08/2021	Suggestions on how to do a literature survey and preliminary investigation on the topic	Many research papers were read , understood and their abstract were to be written.
22/09/2021	Discussion on the implementation of the project	Using Random Forest Classifier we build a machine learning model.
15/10/2021	Discussion on the objective of the project(counting of students at the entrance gate of college in real-time)	Decided to Include the logic of counting students in the program
26/10/2021	Suggestion for counting the number of vehicles like cars, bikes, buses also at the college entrance	Took steps for adding and modifying the program for Login and registration.

10/11/2021	For generation of log files and storing the result, database was advised to be added	Action taken that for each user an entry must be made in the database.
15/11/2021	Discussion on project documentation	Decided to write the content and integrate it in the proper format of the report

Source Code

1. Admin.py

```
from django.contrib import admin
from .models import InsuranceModel

# Register your models here.
@admin.register(InsuranceModel)
class InsuranceModelAdmin(admin.ModelAdmin):
   list_display = ('id', 'prod_info', 'age', 'height', 'weight', 'bmi', 'employee', 'insurance',
'insurancehist1', 'insurancehist2', 'family_his', 'medical_his')
```

2. Apps.py

```
from django.apps import AppConfig


class AppConfig(AppConfig):
        default_auto_field = 'django.db.models.BigAutoField'
        name = 'app'
```

3. Forms.py

```
from django import forms
from .models import InsuranceModel
from django.contrib.auth.forms import UserCreationForm, AuthenticationForm, UsernameField,
PasswordChangeForm, PasswordResetForm, SetPasswordForm
from django.contrib.auth.models import User
from django.utils.translation import gettext, gettext_lazy as _
from django.contrib.auth import password_validation

class CustomerRegistrationForm(UserCreationForm):
   password1 = forms.CharField(label='Password', widget=forms.PasswordInput())
   password2 = forms.CharField(label='Confirm Password (again)',
widget=forms.PasswordInput())
   email = forms.EmailField(label='Email Address', required=True, widget=forms.EmailInput())
   class Meta:
      model = User
      fields = ['username', 'email', 'password1', 'password2']
      labels = {'email': 'Email'}
      widgets = {'username': forms.TextInput()}
class InsuranceForm(forms.ModelForm):
   class Meta:
      model = InsuranceModel
      #fields = ['prod_info', 'age', 'height', 'weight', 'bmi', 'employee', 'insurance', 'insurancehist1',
'insurancehist2', 'family_his', 'medical_his']
      fields = '__all__'
      labels = {'prod_info': 'Product Information','insurancehist1': 'Insurance History 1',
'insurancehist2': 'Insurance History 2', 'family_his': 'Family History','medical_his': 'Medical
History'}
      widgets = {'prod_info': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Product
Info"}),
            'age': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Age"}),
```

```
            'height': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Height"}),
            'weight': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Weight"}),
            'bmi': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Bmi"}),
            'employee': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Employee"}),
            'insurance': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Insurance"}),
            'insurancehist1': forms.TextInput(attrs={'class': 'form-control', 'placeholder':
"Insurance History 1"}),
            'insurancehist2': forms.TextInput(attrs={'class': 'form-control', 'placeholder':
"Insurance History 2"}),
            'family_his': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Family
History"}),
            'medical_his': forms.TextInput(attrs={'class': 'form-control', 'placeholder': "Medical
History"}),
            }
class LoginForm(AuthenticationForm):
  username = UsernameField(widget=forms.TextInput(attrs={'autofocus': True}))
  password = forms.CharField(label=_("Password"), strip=False,
widget=forms.PasswordInput(attrs={'autocomplete': 'current-password'}))

class MyPasswordResetForm(PasswordResetForm):
  email = forms.EmailField(label=_("Email"), max_length=254,
widget=forms.EmailInput(attrs={'autocomplete': 'email', 'class': 'form-control'}))


class MySetPasswordForm(SetPasswordForm):
  new_password1 = forms.CharField(label=_("New Password"), strip=False,
widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}),
help_text=password_validation.password_validators_help_text_html())
  new_password2 = forms.CharField(label=_("Confirm New Password"), strip=False,
widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}))
```

4. Models.py

```
from django.db import models

class InsuranceModel(models.Model):
  prod_info = models.DecimalField(max_digits=20,decimal_places=4)
  age = models.IntegerField()
  height = models.DecimalField(max_digits=20,decimal_places=4)
  weight = models.DecimalField(max_digits=20,decimal_places=4)
  bmi = models.DecimalField(max_digits=20,decimal_places=4)
  employee = models.DecimalField(max_digits=20,decimal_places=4)
  insurance = models.DecimalField(max_digits=20,decimal_places=4)
  insurancehist1 = models.DecimalField(max_digits=20,decimal_places=4)
  insurancehist2= models.DecimalField(max_digits=20,decimal_places=4)
  family_his = models.DecimalField(max_digits=20,decimal_places=4)
  medical_his = models.DecimalField(max_digits=20,decimal_places=4)


  def __str__(self):
    return str(self.id)
```

5. URLS.py

```
from django.urls import path
from . import views
from django.contrib.auth import views as auth_views
from .forms import LoginForm, MyPasswordResetForm, MySetPasswordForm

urlpatterns = [
   path('', views.index, name="index"),
   path('insurance/', views.home,name="home"),
   path('result/', views.result,name="result"),
   path('accounts/login/', auth_views.LoginView.as_view(template_name="app/login2.html",
authentication_form=LoginForm), name='login'),
   path('registration/', views.CustomerRegistrationView.as_view(),
name='customerregistration'),
   path('password_reset/',
auth_views.PasswordResetView.as_view(template_name='app/password_reset.html',
form_class=MyPasswordResetForm), name='password_reset'),
   path('profile/', views.index, name='profile'),
   path('accept/', views.accept, name='accept'),
   path('reject/', views.reject, name='reject'),
   path('logout/', auth_views.LogoutView.as_view(next_page='login'), name='logout'),

]
```

6. Views.py

```
from django.shortcuts import render, redirect
import joblib
from app.forms import InsuranceForm, CustomerRegistrationForm
from django.views import View
from django.contrib import messages

def home(request):
   fm = InsuranceForm()
   return render(request,'app/home.html', {'form': fm})

def index(request):
   return render(request, 'app/index.html')

def result(request):
   if request.method == 'POST':
      cls = joblib.load('finalized_model.sav')
      form = InsuranceForm(request.POST)
      if form.is_valid():
         form.save()
         lis = []
         lis.append(form.cleaned_data['prod_info'])
         lis.append(form.cleaned_data['age'])
         lis.append(form.cleaned_data['height'])
         lis.append(form.cleaned_data['weight'])
         lis.append(form.cleaned_data['bmi'])
         lis.append(form.cleaned_data['employee'])
         lis.append(form.cleaned_data['insurance'])
         lis.append(form.cleaned_data['insurancehist1'])
         lis.append(form.cleaned_data['insurancehist2'])
         lis.append(form.cleaned_data['family_his'])
         lis.append(form.cleaned_data['medical_his'])
```

```
        #ans = cls.predict([[0.076923, 0.059701,  0.600000,  0.131799,  0.272288,  0.000, 6.0,
1.0,   0.000133,  2.0,   5.000000]])
        ans = cls.predict([lis])
        if ans == 0:
            return redirect('/accept')
        else:
            return redirect('/reject')

    else:
        print("empty")
        form = InsuranceForm()
        return render(request, 'app/home.html', {'form': form})

class CustomerRegistrationView(View):
    def get(self, request):
        form = CustomerRegistrationForm()
        return render(request, 'app/registration.html', {'form': form})

    def post(self, request):
        form = CustomerRegistrationForm(request.POST)
        if form.is_valid():
            messages.success(request, 'Congratulations!! Registered Successfully')
            form.save()
        return render(request, 'app/registration.html', {'form': form})

def profile(request):
    return render(request, 'app/profile.html')
def accept(request):
    return render(request, 'app/accept.html')
def reject(request):
    return render(request, 'app/reject.html')
```