International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390269

Car Insurance System

Ms. M. S. Sawalkar [1], Ruturaj Kumbhar [2], Krishna Jamkar [2], Mihir Mandlik [2], Harshawardhan Patil [2]

[*1] Assistant Professor, Department of Computer Engineering, JSPM Narhe Technical Campus, Pune, Maharashtra, India

[2] B.E. in Computer Engineering, JSPM Narhe Technical Campus, Pune, Maharashtra, India

ARTICLEINFO

Article History:

Accepted: 10 April 2023
Published: 30 April 2023

Publication Issue
Volume 10, Issue 2
March-April-2023
Page Number
623-628

ABSTRACT

Many automotive insurance providers are looking to improve their service for their customers, businesses are starting to adapt and implement various methods of analysing data for performance, as a result giving better service for their customers from a better understanding of their needs. The main focus of this project therefore is targeted at automotive insurance providers looking to implement this into their business, the project would also be beneficial to stakeholders. We propose a system built using MERN tech Stack for a better customers experience, by suggesting them the most appropriate cover in time. The requirement for this system is to perform a more efficient up-selling than classic marketing campaigns.

Keywords : Automotive insurance, cover, stakeholders, MERN.

I. INTRODUCTION

In this report, we propose a system built for a better customers' experience, by suggesting them the most appropriate cover in time.

For most famous platforms, such as Amazon and Netflix, users must choose between hundreds or even thousands of products and tend to lose interest very quickly if they cannot make a decision.

Recommendation systems are then essential to give customers the best experience.

Recommender systems are typically employed to support marketing decisions, as they identify statistically validated associations between products and consumers.

These tools have been successfully adopted in many fields; however, not much has been done for the insurance industry.

We are constructing a system for car insurance, to allow agents to optimize up-selling performances, by selecting customers who are most likely to subscribe an additional cover.

The originality of our system is to be suited for the insurance context.

While traditional recommendation systems, designed for online platforms (e.g., e-commerce, videos), are

constructed on huge datasets and aim to suggest the next best offer, insurance products have specific properties which imply that we must adopt a different approach.

Purpose: up-selling. Our goal is to support the agents that are and will continue to be the best advisers for customers, due to their experience and their knowledge of their portfolio.

In short, our tool helps them by automatically selecting from their large portfolios the customers most likely to augment their insurance coverage, in order to optimize up-selling campaigns for instance.

II. THEORY

In order to make good decisions, it is necessary to possess ample amount of information. However, there are several examples showing that too much information is as bad as inadequate information; it is called information overload problem.

Recommender System has been introduced to solve this problem. It is very popular and useful concept in current digital era. It is an information filtering system that suggests products and services most relevant to the User.

Recommender System has been used widely for the products and services, intended for entertainment like music, books, and movies, online games, restaurants and completely based on user ratings.

Recommender systems are typically employed to support marketing decisions, as they identify statistically validated associations between products and consumers.

These tools have been successfully adopted in many fields; however, not much has been done for the insurance industry.

We are constructing a system for car insurance, to allow agents to optimize up-selling performances, by selecting customers who are most likely to subscribe an additional cover. The originality of our system is to be suited for the insurance context.

While traditional recommendation systems, designed for online platforms (e.g., e-commerce, videos), are constructed on huge datasets and aim to suggest the next best offer, insurance products have specific properties which imply that we must adopt a different approach.

Our system combines user generated data and integrates data in order to which customer should be recommended which cover.

Agile Development Model was used as the framework for the development of the system. However, the structure of this thesis is based on Software Development Life Cycle (SDLC) to provide an overview of the project.

A typical Software Development Life Cycle consists of the following stages:

- Planning and requirement analysis: Based on the objective of the product, a project plan will be conducted. The product feasibility will be assessed according to economic resources and the technical requirements. The quality assurance requirements for the product will then be set. Risk assessment is also done in this stage to ensure the project proceeds with minimum risks.

- Defining requirements: Product requirements are to be defined and documented clearly in this stage. This is done through a Software Requirement Specification (SRS) document.

- Designing the product architecture: Base on the SRS document, a Design Document Specification (DDS) document is formed. It contains design proposals for the product architecture including the flow of data with external, and third-party modules if such modules

are used. The document will then be reviewed by important stakeholders to decide on the best approach.

• Developing the product: This is the stage where the product is built. The programming code is generated according to the DDS document. The programming language is chosen depending on which type of software is being developed. Coding guidelines are defined by the developer's organization.

• Testing the product: In this stage, the software is tested for defects. Product defects are reported, tracked and fixed. The testing phase is repeated until the product reaches the standards defined in the SRS.

• Deployment and Maintenance: After the product is carefully tested, it is ready to be deployed. The product is deployed according to the strategy of the organization. Maintenance is done after the product is released when new errors are discovered or when enhancements are needed.

Software Development Life Cycle, or SDLC, widely used in the software industry as a framework to define tasks performed in each step of the development process. The goal of SDLC is to provide the best product with the resources given.

Agile Development Model is one of the two most popular SDLC models, with the other being the Waterfall Development Model.

In opposition to the Waterfall method, the agile method focusses on process adaptability and customer satisfaction by rapid delivery of working software product.

The Agile model is an incremental process model with multiple iterations. Each iteration consists of the fundamental SDLC phases with minor modifications.

These phases are planning, designing, building, testing, reviewing and launching the product. A typical iteration can last from one to three weeks.

Every iteration involves cross-functional teams working simultaneously in their own areas. Agile uses an adaptive approach where there is no detailed planning.

It only clarifies on what feature need to be developed next and what task is required to complete that feature. Development teams can adapt to a product with dynamic requirements.

III.DESIGN APPROACH

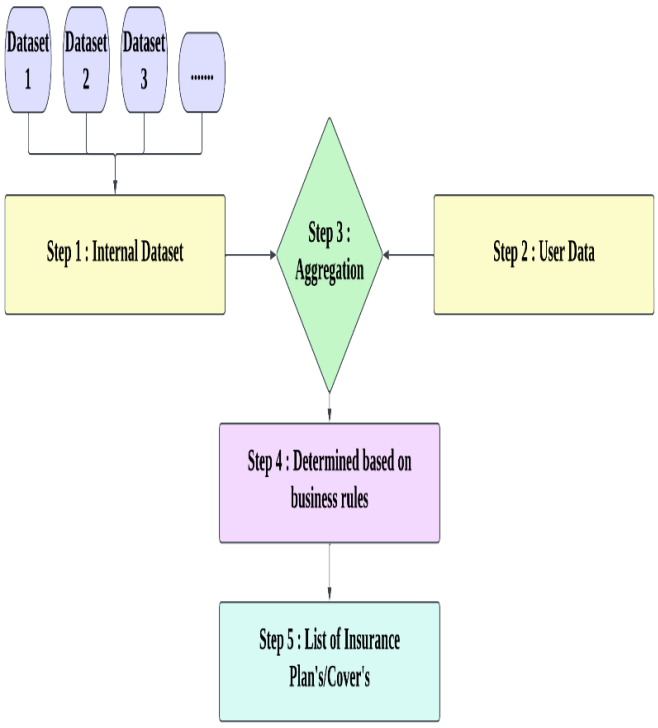

Figure 3: Dataflow Diagram

The model we propose the following approach to build the targeted recommendation system, illustrated by Figure 3.

We built a unique dataset from multiple internal data sources. This dataset includes information about current customers' car policies (current cover, vehicle's characteristics, premium amounts), information about contacts between customers and the insurance company (phone calls, mails, etc.) and claims rate based on customers' history (in particular: claims not covered by their current covering).

This allows us to build relevant features based on existing variables from the initial data. This step is in general based on knowledge of datasets and on intuition supported by experts from specific fields about what could be the most explanatory features.

Next Step we take as an input the customers' data. Its outputs for each customer are estimated by using a token that represents whether a customer added a claim in the past or not, one year after the extraction date of features.

Next Step aims to provide which insurance cover/guarantee is most likely to be added, among the missing covers of the customers. This step answers the question: which additional insurance cover should we recommend?

For instance, some customers are already covered by recommended cover because they subscribed to an old version of car insurance product, whose guarantees were defined differently. Some simple business rules downstream of the model take into account these particularities.

Last Step is list of insurance covers which can be claimed by the user according his/her needs.

IV. RESULTS AND DISCUSSION

This thesis details the development process of building an application that serves as one of these systems. The application is based on ReactJS.

All concepts and technologies used in the project are explained in their corresponding sections.

The result of this project was a web-based application that can be used for Car Insurance Recommendation.

The system acts as a centralized control panel and enables the user to manage his/her car's data.

The main method of development applied in this project was Agile methodology as it is suitable for the changing requirements of the project.

The Agile is an incremental model, consisting of many iterations. However, this thesis structure follows the traditional Software Development Life Cycle stages to provide an overview of the entire development process.

The overall aims of this project were to produce a claim a new policy holder would produce an insurance claim in the next year.

This is inthe hopes that it can make automotive insurance more accessible to more drivers, through a solution that produces a more accurate cover.

To make claim it would have to effectively apply the feature information such as the customer's location or region, as well as their personal and vehicle information.

While it may havebeen slightly difficult to directly reference real-world features that were anonymized, this was possible to overcome by analyzing feature's importance through user generated data.

Using user generated data, we are able to produce a solution that is more complex than a simple one, and provide reliable claim cover.

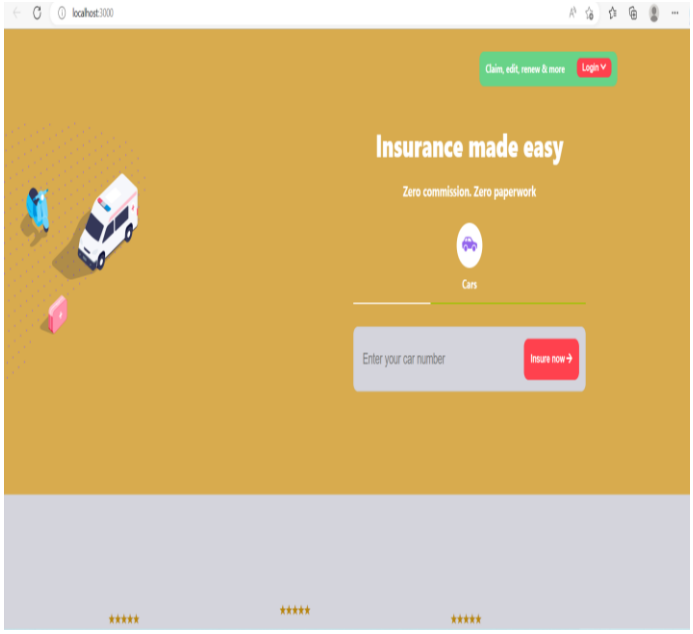

Figure 4.1 : Homepage of website.

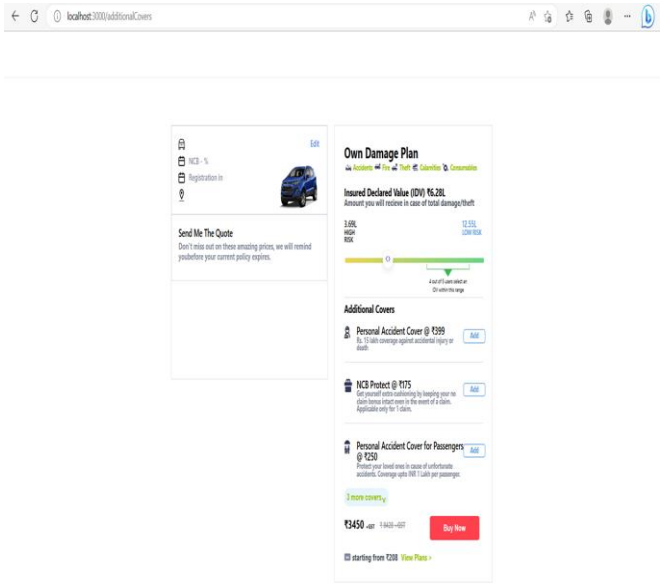

Figure 4.2 : Insurance plan covers provided to the user.

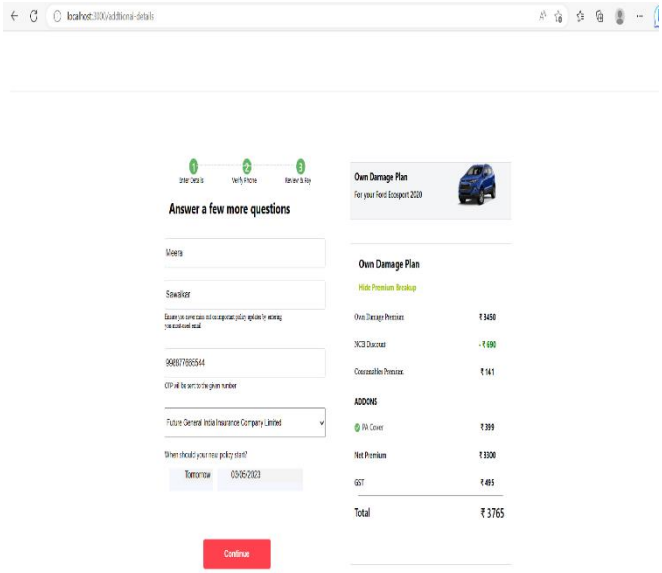

Figure 4.3 : Final cover plan details.

All concepts and technologies used in the project are explained in their corresponding sections. This includes the relevant theoretical background such as client-server model, Content Management System, and Document Object Model.

The result of this project was a web-based application that can be used for Car Insurance Recommendation.

The system acts as a centralized control panel and enables the user to manage his/her car's data.

The main method of development applied in this project was Agile methodology as it is suitable for the changing requirements of the project.

The Agile is an incremental model, consisting of many iterations. However, this thesis structure follows the traditional Software Development Life Cycle stages to provide an overview of the entire development process.

The overall aims of this project were to produce a claim a new policy holder would produce an insurance claim in the next year.

This is in the hopes that it can make automotive insurance more accessible to more drivers, through a solution that produces a more accurate cover.

To make claim it would have to effectively apply the feature information such as the customer's location or region, as well as their personal and vehicle information.

While it may havebeen slightly difficult to directly reference real-world features that were anonymized, this was possible to overcome by analyzing feature's importance through user generated data.

Using user generated data, we are able to produce a solution that is more complex than a simple one, and provide reliable claim cover.

V. CONCLUSION

In conclusion, the Car Insurance System (CIS) project was a success.

The CIS achieved all the predefined functions and quality requirements. With each implementation, there was considerable effort to enhance extensibility.

As a result, the product had high maintainability, modifiability, and scalability.

Deployment and maintenance will be done in the upcoming future and is outside of the scope of this paper.

While the CIS is satisfied with the final result, there were still many aspects that can be optimized.

In our opinion, there are always potential for improvements in code structure, functional logic, user experience design and application performance.

We will continue to do code refactoring to improve readability, maintainability and reduce complexity.

VI. FUTURE WORK

From carrying out this project we were intrigued by the results we had obtained from at various different stages.

Due to the short time given however, we did not explore in depth all the aspects of desired project we would have liked to.

In future we would like to implement the same using machine learning algorithms and train unsupervisedand semi-supervised models, to determine whether or not their results could be used to improve the insurance claim predictions.

There are also certain aspects we would go further in depth for researching and selecting, such as the choice user generated data.

The addition of more modules for generating more data would also be a good way to improve on the project in future, because they can yield great results on various projects.

It would also be a good idea to train a live system (perhaps a batch system) which takes in batches

of new customer information, and then produces claims automatically within a reasonable time.

With a live system the data can thereby increase with more customer information and this could potentially lead to a greater number of claims if there is more data it can also be used for review of the system.

Lastly due to the fact the data was anonymized we were unable to make discoveries about the meaning in its real-world applications, we were only able to make claims with regards to common cover statistics.

If there had been more time we would have liked to try and specifically cross-reference feature values to certain statistics with more research, for example : Life insurance or bike insurance regards to vehicle and producing claims.

VII. REFERENCES

[1]. Bass, L., Clements, P. & Kazman, R. 2003. Software Architecture in Practice. 2nd ed. Boston: Addison Wesley.
[2]. Crockford, D. 2008. JavaScript: The Good Parts. 1st ed. O'Reilly Media: O'Reilly
[3]. Flanagan, D. 2011. JavaScript: The Definitive Guide. 6th ed. Fonenix: O'Reilly
[4]. Mauthe, A. & Thomas, P. 2005. Professional Content Management Systems: Handling Digital Media Assets. 2nd ed. John Wiley
[5]. GitHub. 2017. Usage. Webpack. Accessed 2018. https://github.com/webpack/docs/wiki/usage
[6]. Lindley, C. 2014. Front-end Driven Applications-A New Approach to Applications. Telerik. Accessed 2018. https://developer.telerik.com/featured/front-end-driven-applications-new-approach-applications/
[7]. React. No date. Accessed 2018. https://reactjs.org/
[8]. Bobriakov, Igor. (2018). Top 10 Data Science Use Cases in Insurance. Accessed 12/02/2020, from https://medium.com/activewizards-machine-learning- company/top-10-data-science-use-cases-in-insurance-8cade8a13ee1
[9]. React. No date. Introducing JSX. Accessed 2018. https://reactjs.org/docs/introducingjsx.html
[10]. React. No date. React-Component. Accessed 2018. https://reactjs.org/docs/react-component.html
[11]. W3C. 2016. Cascading Style Sheets. Accessed 2018. https://www.w3.org/Style/CSS/
[12]. W3C. 2017. HTML 5.2. Accessed 2018. https://www.w3.org/TR/html52/

Cite this article as :
M. S. Sawalkar, Ruturaj Kumbhar, Krishna Jamkar, Mihir Mandlik, Harshawardhan Patil, "Car Insurance System", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 2, pp.623-628, March-April-2023. Available at doi : https://doi.org/10.32628/CSEIT2390269
Journal URL : https://ijsrcseit.com/CSEIT2390269