

Customer Lifetime Value Prediction for Auto-Insurance Companies

Nithin Goud Kumbam, Rindhuja Treesa Johnson, and Prudhvi Yaswanth Mundluri

University of Maryland - Baltimore County

DATA 606 - Capstone Project

Jester Ugalde

May 15, 2024

Customer Lifetime Value Prediction for Auto-Insurance Companies

Introduction

Customer lifetime value (CLV) is a business metric used to determine the amount of money customers will spend on your products or services over time. (Danao, 2023). Insurance companies follow specific business models that involve assuming and diversifying risk. Each customer insured using the company's plan is a risk. The company pools the individual risks and then uses the premium collected from these individual clients to invest in high-interest-generating assets like bonds.(Ross, 2021) The success of an insurance company, is, therefore, determined mainly by two factors - the client premium and the investment return.

As a result, it is evident that the value of a customer in different dimensions (premium amount, duration of the plan, claims, claim frequency, and more) is relevant in determining the profit of the insurance firm. In this project, we analyzed the different variables that contribute to the value of a customer that paved the way to modeling a machine learning model that predicts the Customer Lifetime Value for new clients based on their unique characteristics. Further, we developed and hosted a website where the company can easily find the CLV for an incoming client based on the model. We also designed a dashboard that gives a glimpse of the current state of insurance clients for a bigger picture. Finally, we implemented a 'Q&A' interface using Google's Gemini LLM that converts human questions to SQL queries and retrieves data from our database.

Business Use

An Auto Insurance Company X in the USA is experiencing severe customer churn and wants to advertise promotional offers to retain its loyal customers. As the Data Science Team for company X, we analyze and understand the different parameters involved in the business. It's a competitive market for insurance companies, and the insurance premium is not the only determining factor in a client's choices. CLV is a customer-centric metric and a powerful base to build upon to retain valuable customers, increase revenue from less valuable customers, and improve the customer experience overall. Considering the above reasons, The team agreed to use Customer Lifetime Value CLV as the parameter for evaluating customer reliability.

Table 1*The Columns in the dataset*

Number	Name	Count	NULL/non-NULL	Type
0	Customer	9134	non-null	object
1	State	9134	non-null	object
2	Customer Lifetime Value	9134	non-null	float64
3	Response	9134	non-null	object
4	Coverage	9134	non-null	object
5	Education	9134	non-null	object
6	Effective To Date	9134	non-null	object
7	EmploymentStatus	9134	non-null	object
8	Gender	9134	non-null	object
9	Income	9134	non-null	int64
10	Location Code	9134	non-null	object
11	Marital Status	9134	non-null	object
12	Monthly Premium Auto	9134	non-null	int64
13	Months Since Last Claim	9134	non-null	int64
14	Months Since Policy Inception	9134	non-null	int64
15	Number of Open Complaints	9134	non-null	int64
16	Number of Policies	9134	non-null	int64
17	Policy Type	9134	non-null	object
18	Policy	9134	non-null	object
19	Renew Offer Type	9134	non-null	object
20	Sales Channel	9134	non-null	object
21	Total Claim Amount	9134	non-null	float64
22	Vehicle Class	9134	non-null	object
23	Vehicle Size	9134	non-null	object

In Company X's terms, Customer Lifetime Value represents a client's value to company X over a period of time. Using CLV effectively can improve customer acquisition and customer retention, prevent churn, help the company to plan its marketing budget, measure the performance of its ads in more detail, and much more.

Data Summary and Extraction

For this business case, we extracted open data available on Kaggle under the competition - IBM Watson Marketing Customer Value Data. The dataset consists of insurance policy data of 9134 unique auto insurance clients in company X as of the year 2011. Each policy is defined by 24 attributes including the Customer Lifetime Value for each client. The dataset is uploaded on an AWS S3 bucket and interacted using AWS API for Python.

Table 1 gives a glimpse into the data attributes in the dataset, its count, the presence of any NULL values, and the data type. We have the data of insurers from 5 states in North America - Washington, California, Arizona, Nevada, and Oregon. The data consists of 8 continuous variables and 15 categorical variables which are one-hot encoded for applying ML models. Among the variables, *Customer* (unique identifier) and *Effective To Date* are discarded as they are irrelevant to our analysis.

Data Cleaning and Pre-processing

The dataset **do not** have any missing values.

The dataset was divided into two based on the data types. the integer and Float types into a numerical dataset and the object types into a categorical dataset. The Categorical variables were one-hot encoded later on.

Besides, we checked for outliers within the dataset, however, after analyzing the box plots of numerical variables, we concluded to keep all the data points as each of them have unique characteristics based on 22 different attributes and each of them could be relevant to our analysis and modeling.

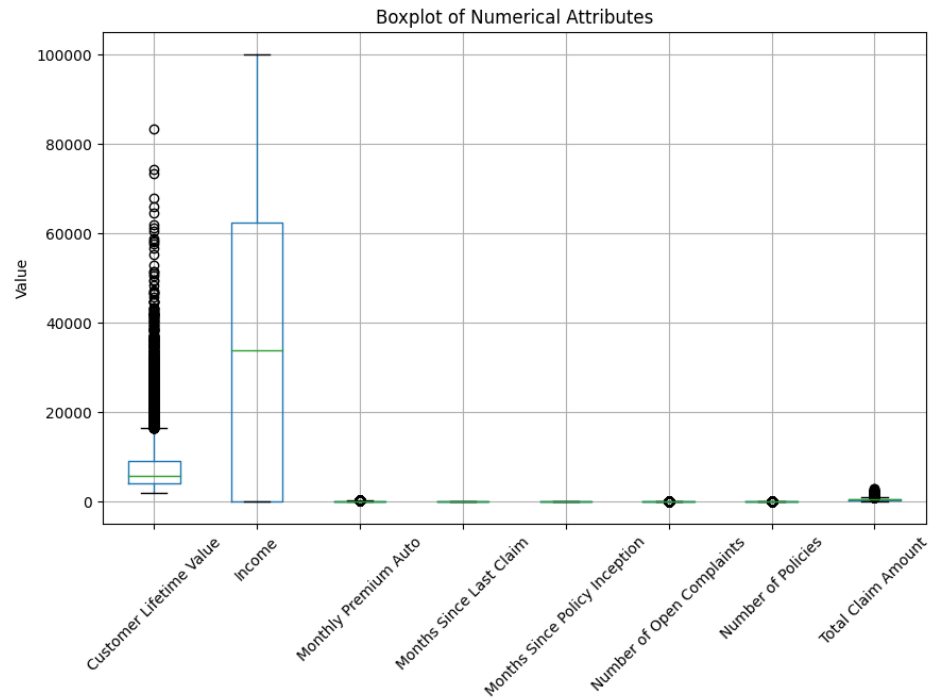


Figure 1

Box plot for the numerical attributes in the dataset showing the quartiles and outliers

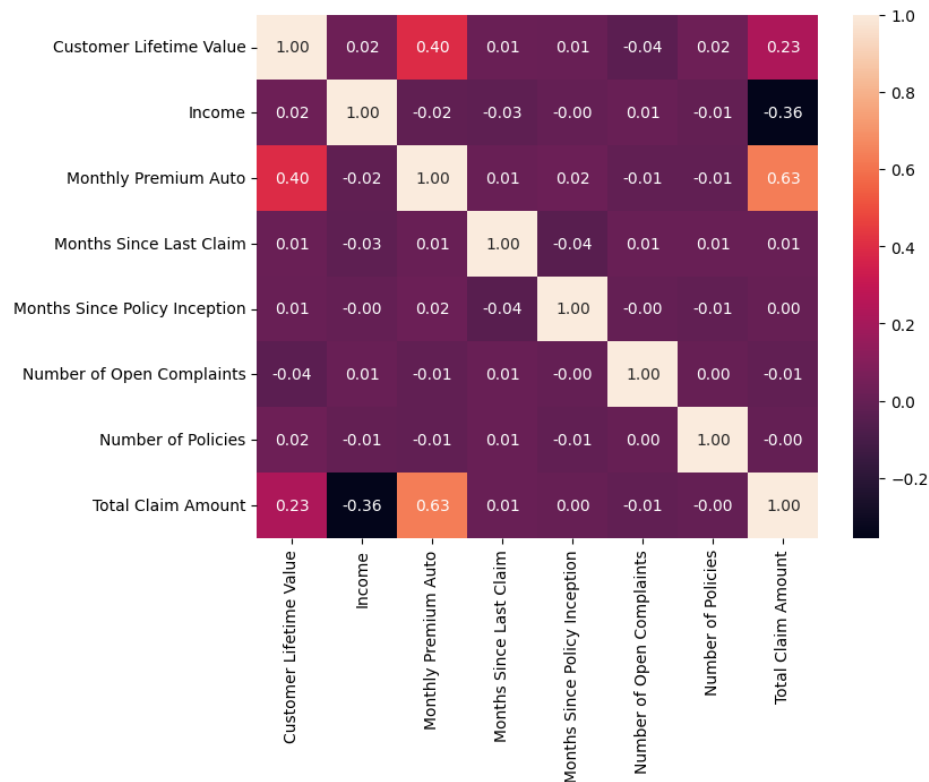
Figure 1 shows the box plot of the numerical attributes in the dataset. It can be interpreted that none of the training variables have a considerable outlier. The outliers, in this case, cannot be neglected or replaced because each of them shows a unique customer behavior and is important in determining exceptional clients in model training.

Exploratory Data Analysis and Visualization

We conducted an in-depth analysis of the given dataset using various methods. Our primary focus throughout the analysis was to determine the most relevant variables that affect the customer lifetime value, which will be referred to as the target variable from now on. The following analysis methods were implemented.

Correlation and Scatter plots

Figure 2 verifies that none of the variables are correlated and thereby reduces the risk of multicollinearity. Further, the scatter plot grid in Figure 3 verifies that the data points are randomly distributed between any two pairs of variables, proving the absence of any collinearity.

**Figure 2**

The Pearson Correlation Heat map between each of the numerical variables

Uni-variate Analysis - Histograms and Bar graphs

We implemented Histograms for numerical variables and bar graphs for the categorical variables. This helped us understand the distribution of different variables in the dataset.¹ We observed that the average income of the population is skewed to the left due to the dominance of clients with zero income. Therefore, we categorized the income variable along with a few other variables and the target variable to try the segmentation of the clients.

Figure 4 shows that there are over 2300 clients out of 9134 subjects who have a zero income and the rest of the incomes are almost uniformly distributed, except for a bump from 20k to 30k. Also, the majority of the clients have a single automobile insurance plan.

Figure 5 again confirms that the majority of the clients have a lower income range and

¹ Refer the Project GitHub Repository for the complete analysis and plots.

have a CLV between 4000-8000.

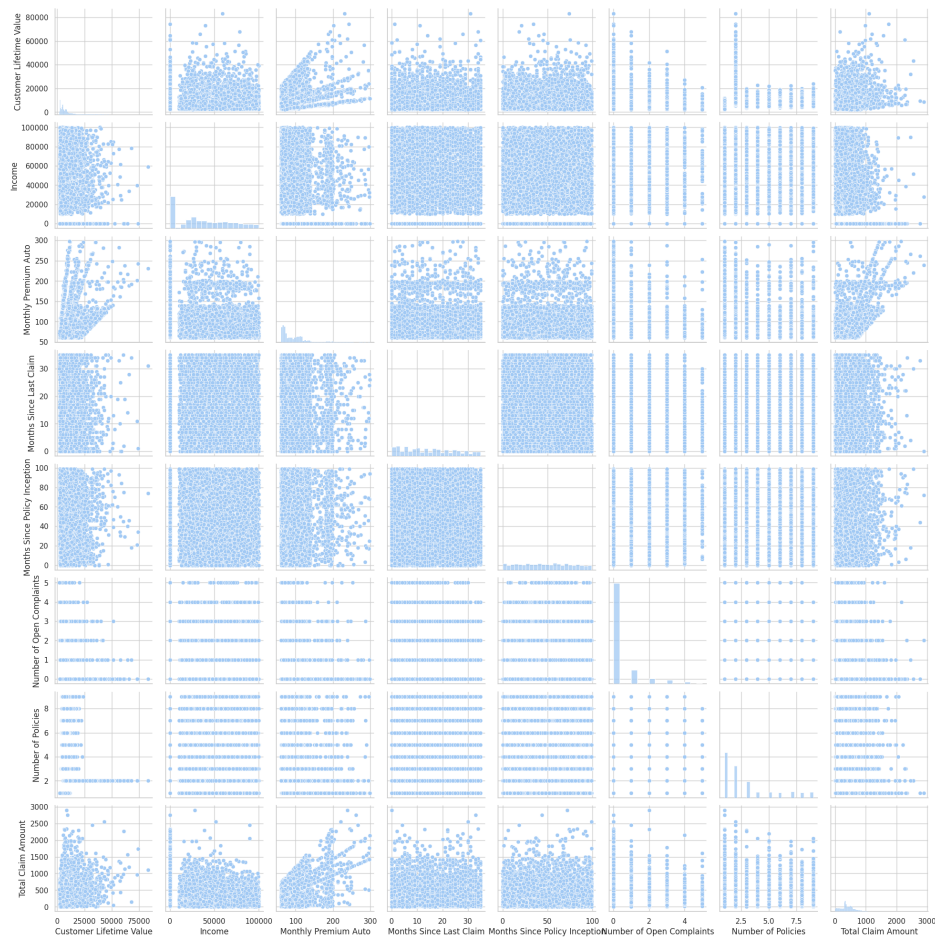


Figure 3

Scatter plot between all pairs of variables

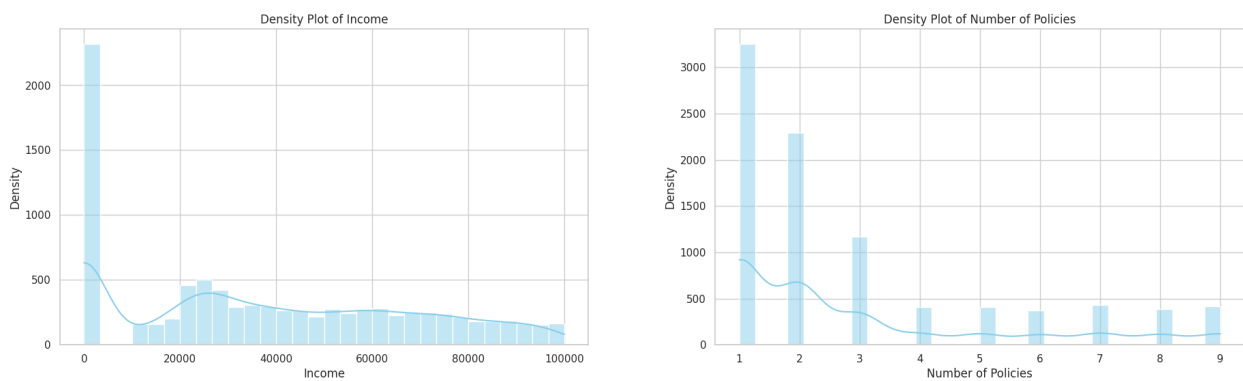
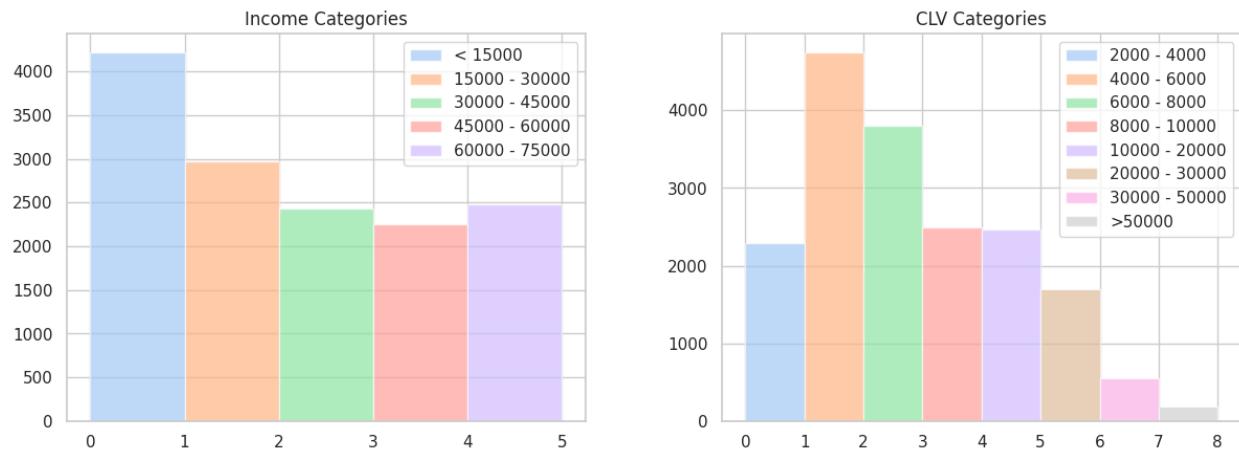
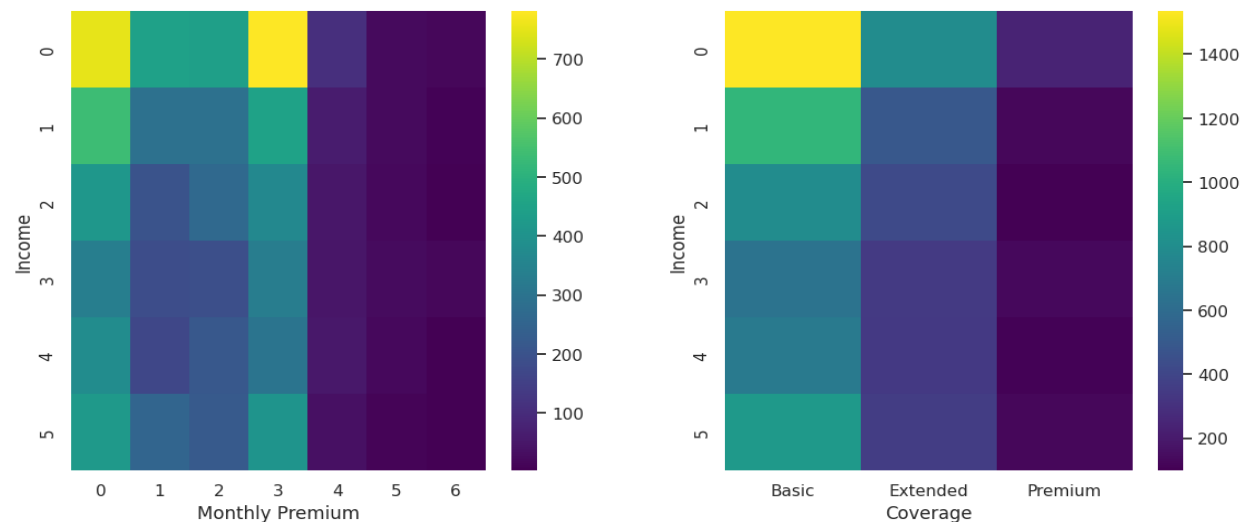


Figure 4

Example of Dense Plots for continuous and categorical numerical variables

**Figure 5**

Example of Bar plots of continuous variables after binning into categories of custom sizes

**Figure 6**

Heat map of Income vs (a) Monthly Premium Amount (b) Coverage Plan grouped by the categories

Bi-variate Analysis

We implemented *Group By* clause to understand how each pair of variables behaved. We utilized the categorized data of the numerical continuous variables to make this analysis meaningful.

Figure 6 gives us the insight that even though we might expect the lower income class (scale ascending) to have a lower premium amount and vice-versa, that is not the case. Despite the income, the majority of the clients tend to have an above average monthly premium amount (> \$400). Moreover, regardless of the income, the clients prefer to choose the Basic Insurance plan.

Dimensionality Reduction and Clustering

Throughout the analysis, we tried to pick up the essential variables that determine the target variable. We used the Principal Component Analysis and t-distributed Stochastic Neighbor (t-SNE) methods to capture the relevant data.

From the figure 7, we can see that PCA defined extracted variables beyond the dense area of the actual data points. On the other hand, t-SNE failed to extract all the features of the dataset as the data points are uniformly spread in the plane.

Figure 8 clearly conveys that segmentation using K-Means clustering is not an approach for this dataset.

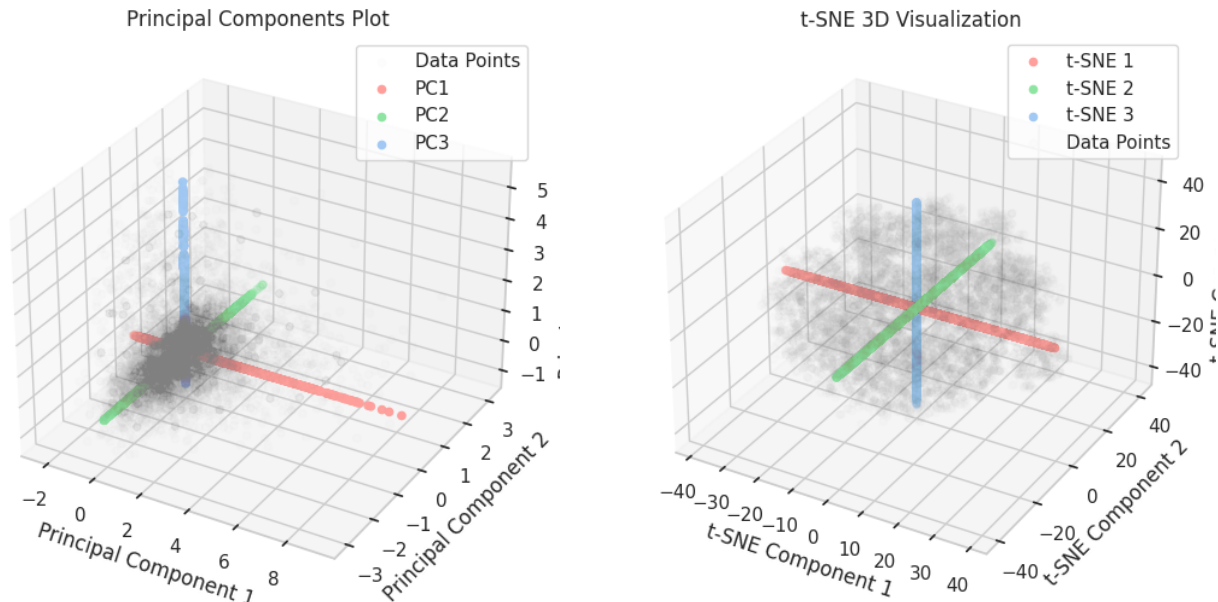
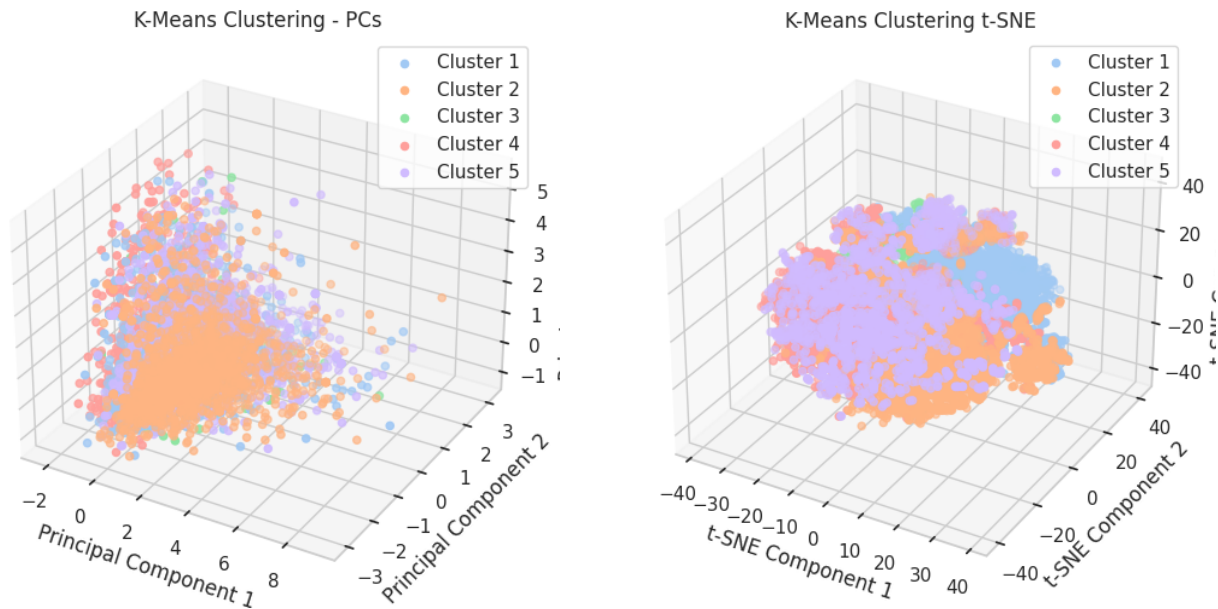


Figure 7

Three significant variables formed using (a) PCA (b) t-SNE from the data points

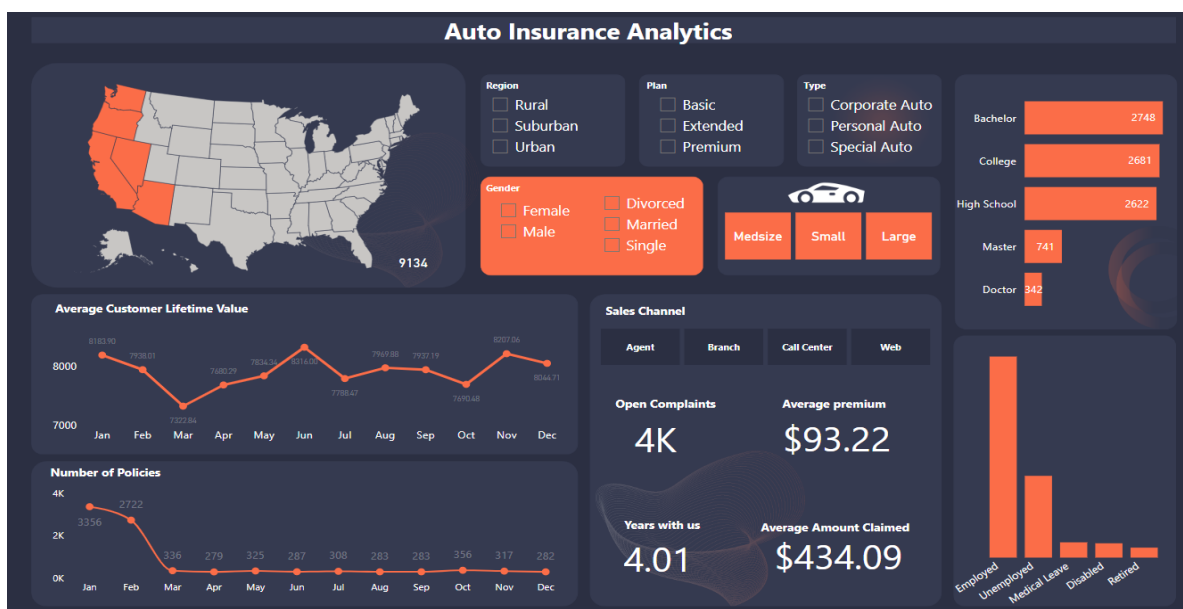
**Figure 8**

Attempting to segment Customers into 5 clusters using K-Means Clustering (a) PCA variables (b) *t*-SNE variables

Dashboard

Figure 9

Interactive Dashboard using MS Power BI



Using the dataset, we developed an interactive MS Power BI dashboard that can be considered a one-stop destination for any data-related concerns. We implemented 9 numerous filters to enable customized visualization of how Customer Lifetime Value is decided based on characteristics like State, Gender, Plan, Car size, and more. ²

Machine Learning Models and Optimization

Data Preparation

The core of this project is to develop a machine-learning model that best predicts the target variable for a new client. It is crucial to prepare the data for modeling and the main steps for preparation were -

- Standardizing identifiers: To avoid any errors or warnings while modeling, we standardized the column names of the dataset by removing the spaces and adding underscores instead.
- Encoding the Categorical variables: All the ML models require numerical data to process and therefore, we encoded all the object type variables using the *LabelEncoder()* function.
- Splitting the Dataset: After removing the target variable from the dataset, the data frame was split into train and test sets with a test size of 30%.

The data was prepared for modeling!

Machine Learning

Linear Regression

For Linear Regression, we utilized the Lasso (L1) and Ridge (l2) Regularizations to optimize the performance metrics of the model by avoiding over-fitting.

² Check out the dashboard

Table 2*Linear Regression: Performance Metrics*

Model/Metric	RMSE	R-Squared (Train)	R-Squared (Test)
Lasso (L1)	0.5993	0.1950	0.1968
Ridge (L2)	0.5811	0.2498	0.2449

Decision Tree Regressor

On modeling the data with the Decision Tree Regressor, we encounter an R-squared of 1.0 on the train data and only 0.84 on the test data indicating an over-fitting of the model on the train dataset.

Table 3*Decision Tree Regressor: Performance Metrics*

Model/Metric	RMSE	R-Squared (Train)	R-Squared (Test)
Decision Tree	0.2608	1.0	0.8479

Random Forest Regressor

With the Random Forest Regressor, we obtain a train data accuracy (R^2) of 98% and test data accuracy of 90%. Among all the models until this point, Random Forest Regressor proves to be the most reliable prediction model.

Table 4*Random Forest Regressor: Performance Metrics*

Model/Metric	RMSE	R-Squared (Train)	R-Squared (Test)
Random Forest	0.2056	0.9829	0.9054

Random Forest Regressor with Hyper-parameter Tuning

We implemented Grid Search cross-validation on a set of random forest parameters like maximum depth, number of estimators, bootstrap, and maximum features to train the model using

tuned hyper-parameters to improve the test accuracy. Even though the test accuracy increased by a percent, we concluded that it was not worth the computing time and efficiency to include that in the model. Therefore, despite the better performance, we discarded the hyper-parameter tuning on the RF regressor.

Table 5

Hyper-parameters tuned RF: Performance Metrics

Model/Metric	RMSE	R-Squared (Tuned - Test)	R-Squared (non-Tuned - Test)
Tuned RF	0.1967	0.9134	0.9054

Adaptive Boosting (AdaBoost) with Random Forest Regressor

To further investigate the fitting of the Random Forest Regressor, we adapted the Adaptive Boosting algorithm. We used an ensemble of decision trees or the random forest model as the base estimator that fits the train data with the parameters. AdaBoost picks the weak learners, combines them, gives them additional weightage, and makes a strong regressor. Finally, the prediction is made by combining the predictions of all weak models, with each model's contribution weighted according to its accuracy. However, the test accuracy has come down as compared to the base RF model and therefore, we discarded it.

Table 6

AdaBoost with RF: Performance Metrics

Model/Metric	RMSE	R-Squared (AdaBoost - Test)	R-Squared (RF - Test)
AdaBoost	0.2171	0.8946	0.9054

Neural Network

We proceeded to the deep learning libraries, TensorFlow and Keras, by implementing Linear and Rectified Linear Unit activation layers along with the Adam optimizer with 100 epochs to model the train set. Although the computational process was more extensive, the

resulting accuracy fell short of that achieved by the basic Random Forest model, thus rendering it unsuitable for prediction purposes.

Table 7

Neural Network: Performance Metrics

Model/Metric	R-Squared (Neural - Test)	R-Squared (RF - Test)
Neural Network	0.8797	0.9054

Summary - ML models

Upon the completion of modeling and testing the data with the above models - Tables 2, 3, 4, 5, 6, and 7 - we decided to use the basic Random Forest Regressor model to predict new Customer Lifetime Values due to its better performance and lesser computational requirements.

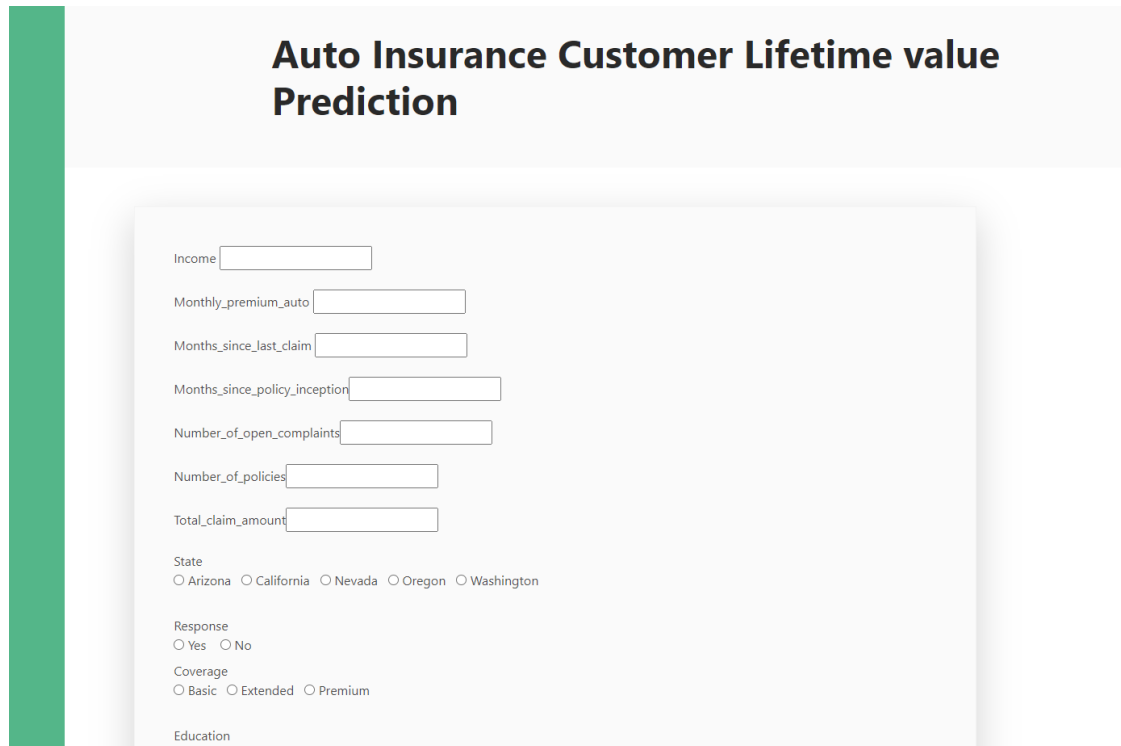
User Interface for CLV Predictions

For Company X to understand customer behavior before issuing an insurance policy, we developed a website scripted in HTML and styled by CSS and JavaScript that gives an instant prediction of the Customer Lifetime Value of any potential client taking into consideration the characteristics we utilized for the prediction model.

We converted the Random Forest prediction model into a .sav file and connected it to an HTML file using the Python libraries *pickle* (for connecting to the .sav file and executing) and *flask* (rendering the webpage upon running the model).

Figure 10 is the landing page of the website that predicts the Customer Lifetime Value for any unique customer given all the 19 attributes required to predict it. This gives Company X an upper hand in deciding the risk involved with each of their incoming clients thus lowering the chances of loss. ³

³ The website is not rendered publicly due to charges involved in securing a deal with a web server. The scripts and files can be seen in the GitHub Repository



The screenshot shows a web form titled "Auto Insurance Customer Lifetime value Prediction". The form contains several input fields and radio button groups. The fields are: "Income", "Monthly_premium_auto", "Months_since_last_claim", "Months_since_policy_inception", "Number_of_open_complaints", "Number_of_policies", and "Total_claim_amount". Below these are three radio button groups: "State" with options "Arizona", "California", "Nevada", "Oregon", and "Washington"; "Response" with options "Yes" and "No"; and "Coverage" with options "Basic", "Extended", and "Premium". At the bottom, there is an "Education" label followed by a series of dashes.

Figure 10

A Glimpse of the web page designed for CLV Prediction

Q&A Interface for Data Retrieval with LLM

Further, we extended this project to implement a Large Language Model for extracting data from the database. We obtained the Python API keys for applying Google's Gemini LLM model using the *GooglePalm* class from the *langchain* library (LangChain, 2024a). The LLM class object is used to convert human prompts to SQL queries.

The Python interface reads the data-related question in human language, passes it to the *langchain* object that converts it to machine-readable language, passes it to *SQLDatabaseChain* object that converts it into a SQL query, then the *SQLDatabase* (LangChain, 2024b) framework connects MySQL Server to the Python application and executes the query by retrieving the desired data.

Thus, we can get any information (Figure 11) from the database based on the data attributes listed on the page for reference. Any questions beyond the scope of the data will throw

an error.

Auto insurance: Database Q&A

You can ask questions using the column names: Customer, State, Customer Lifetime Value , Response , Coverage ,Education , Effective To Date , EmploymentStatus , Gender ,Income , Location Code , Marital Status , Monthly Premium Auto ,Months Since Last Claim , Months Since Policy Inception ,Number of Open Complaints , Number of Policies , Policy Type , Policy , Renew Offer Type , Sales Channel , Total Claim Amount ,Vehicle Class , Vehicle Size

Question:

how many states data is available and name them?

Answer

5, Washington, Arizona, Nevada, California, Oregon

Figure 11

A Glimpse of the Q&A page designed for Data Retrieval using LLM and MySQL Database

Results

Table 8

Performance Metrics of ML Models

Model/Metric	RMSE	R-Squared (Train)	R-Squared (Test)
Lasso (L1)	0.5993	0.1950	0.1968
Ridge (L2)	0.5811	0.2498	0.2449
Decision Tree	0.2608	1.0	0.8479
Random Forest	0.2056	0.9829	0.9054
Tuned RF	0.1967	-	0.9134
AdaBoost	0.2171	-	0.8946
Neural Network	-	-	0.8797

From the table 8, it is observed that the tree-based models outperform other models. The Hyper-tuned Random Forest regressor gives the best test data accuracy followed by the basic Random Forest regressor. However, the computational efficiency outweighs the small error margin and therefore, we picked the Random Forest Regressor as our go-to model for the prediction of the CLV for Company X.

Besides the model, the extension of the project to develop a user-friendly web page to enable the live prediction of Customer Lifetime values by the Company X customer recruitment staff gives the company the advantage of seeing through their customers even before adding in the policy.

The Q&A interface enables the Data Science team to further investigate the data and retrieve insights from time to time with updates from new and existing customers.

Summary

This Data Science Project leveraged by Company X for the prediction of Customer Lifetime Value of new customers gained the following outcomes -

- **Improved Customer Retention:** Identified high-value customers for targeted promotional offers and loyalty programs, leading to increased customer retention and reduced churn rates.
- **Enhanced Marketing Effectiveness:** Enabled data-driven allocation of marketing resources towards high-value customer segments, maximizing return on investment.
- **Data-driven Decision Making:** Empowered stakeholders with CLV insights and interactive visualizations, facilitating informed decisions regarding customer acquisition, retention, and overall business strategy.
- **Enhanced User Experience:** Provided a user-friendly Q&A interface for easy access to information, promoting data democratization and knowledge sharing within the organization reducing the time required for developing efficient SQL queries.

References

Danao, M. (2023). What is customer lifetime value (clv). *Forbes Advisor*.

<https://www.forbes.com/advisor/business/customer-lifetime-value/#:~:text=What%20is%20customer%20lifetime%20value%20and%20why%20is%20it%20important,and%20increase%20profits%20over%20time>

LangChain, I. (2024a). Llms. <https://python.langchain.com/docs/integrations/llms/>

LangChain, I. (2024b). Sql database.

https://python.langchain.com/docs/integrations/tools/sql_database/

Ross, S. (2021). How do insurance companies make money? business model explained.

Investopedia. <https://www.investopedia.com/ask/answers/052015/what-main-business-model-insurance-companies.asp#:~:text=The%20essential%20insurance%20model%20involves,into%20other%20interest%2Dgenerating%20assets.>