Understanding The Customer Value of IBM Watson

STA6704 - FINAL PROJECT REPORT

Yassin Khudari  
Master of Science in Data Analytics *University of Central Florida*  
Orlando, Florida, USA, *mohammadyassin.khudari@ucf.edu*

Amanda Consiglio   
Master of Science in Data Analytics *University of Central Florida*  
Orlando, Florida, USA, *aconsiglio@knights.ucf.edu*

Jainam Shah  
Master of Science in Data Analytics *University of Central Florida*  
Orlando, Florida, USA *jainam.shah@ucf.edu*Samuel Soares Walton  
Master of Science in Data Analytics  
*University of Central Florida  
Orlando,Florida, USA  
samsoareswalton@knights.ucf.edu*

***Abstract*—**Understanding customer behavior and predicting customer lifetime value is crucial for businesses seeking to improve customer retention and profitability. The IBM Customer Value dataset provides a comprehensive set of customer attributes that allow us to gain insights into customer demographics and interactions. This report/paper explores the prediction of customer lifetime value (CLV) using the IBM Customer Value dataset for Watson Analytics. The dataset provides relevant customer information, demographics, and buying behavior, allowing us to develop targeted customer retention programs through predictive analytics. The primary goal is to analyze customer data and understand how to retain the most profitable customers and increase their response, retention, and growth. In this study, we perform exploratory data analysis (EDA) on the dataset and consider the suitability of Principal Component Analysis (PCA) for dimensionality reduction. Efficient PCA results were obtained which were then utilized to build models using various predictive modeling techniques, including Generalized Linear Models (GLM), K-Means clustering, Elastic Net Regression (Lasso and Ridge), and tree-based methods such as Decision Trees, Random Forest, and Boosted Trees.

***Keywords—Principal Component Analysis (PCA), EDA, Tree-Based Methods, Generalized Linear Model (GLM), Elastic-Net Regression, tree-based Methods***

# Introduction

In the modern business landscape, understanding and predicting customer behavior is of paramount importance for effective marketing strategies and business growth. The concept of Customer Lifetime Value (CLV) has emerged as a crucial metric, providing valuable insights into the long-term profitability of customer relationships. Accurate prediction of CLV enables businesses to optimize resource allocation, tailor marketing efforts, and enhance overall customer satisfaction. The dataset, carefully curated and prepared, comprises an extensive array of customer-related attributes, offering invaluable insights into the behaviors and characteristics of customers. The primary objective of this study is to build robust and accurate predictive models for estimating Customer Lifetime Value. In this report, we present an in-depth exploration of Customer Lifetime Value prediction using advanced modeling techniques on the IBM Watson Marketing Customer Value Data.

Our comprehensive approach involves four main stages: data cleaning, exploratory data analysis (EDA), Principal Component Analysis (PCA),and the application of multiple sophisticated modeling techniques such as Generalized Linear Model (GLM), k-means clustering, tree-based analysis (Decision Tree, Random Forest, and Boosted Tree), and Elastic Net regression (Lasso and Ridge).

High-quality data is the foundation for reliable predictive modeling. In this phase, we address data inconsistencies, missing values, outliers, and other data quality issues that may impact the integrity of our analyses. Through meticulous data cleaning, we aim to ensure that our models are built on accurate and reliable data, enhancing the validity and robustness of our predictions.

EDA is a crucial step to gain valuable insights into the dataset and understand the underlying patterns, trends, and relationships between variables. We employ various data visualization techniques, including bar plots, box plots, and correlation matrices, to uncover meaningful information and potential features relevant for CLV prediction. EDA guides our feature selection process, assisting us in identifying the most influential variables for modeling.

The high dimensionality of modern datasets can lead to computational challenges and potential overfitting when building predictive models. To address this issue, we employ PCA, a dimensionality reduction technique, to transform the original features into a new set of orthogonal components. By retaining the most informative components while reducing noise and redundancy, PCA enables us to create more efficient models without compromising predictive accuracy.

We explored an array of sophisticated modeling techniques to predict CLV accurately. We begin with the Generalized Linear Model (GLM) to establish a baseline and understand the linear relationship between dependent and independent variables. We then leverage k-means clustering to segment customers based on similar

The rest of the paper is organized as follows: Section 2 presents data cleaning procedure. Section 3 describes the exploratory data analysis of the dataset. Section 4 presents the principal component analysis used for dimensionality reduction, Section 5 presents Model analysis based on different regression and tree-based models built for accurate analysis and prediction, Section 6 shows the results of models interpreted through RMSE values, Section 7 represents the technical conclusion of paper and Finally, Section 8 shows the executive summary for better understanding of stakeholders, summarizing the key findings and implications for businesses.

# Data Cleaning

Before commencing, we cleaned the data by first checking for null values for each of the variables. The dataset did not contain any null values and none of the variables looked like they could cause any target leakage. We then examined both the categorical and numeric variables to see if there were any observations that appeared out of place. This was done by looking at the mean, minimum, maximum, and quartiles of the numeric values, in addition to the label counts of the categorical. From both of these we concluded that none of the values were suspicious. Next, we plotted all of the numeric variables to see their distributions and found that many of them had a positive skew, including the target variable. We also dropped the customer column, since it is a column of unique identifiers for each of the customers and does not provide any additional information. A label encoder function was used to turn the labels into numeric values for the categorical variables. Lastly, Effective to Date was dropped from the dataset since we did not want to turn our data into a time series which could overcomplicate or not be suitable for a GLM We also removed any outliers in the data.

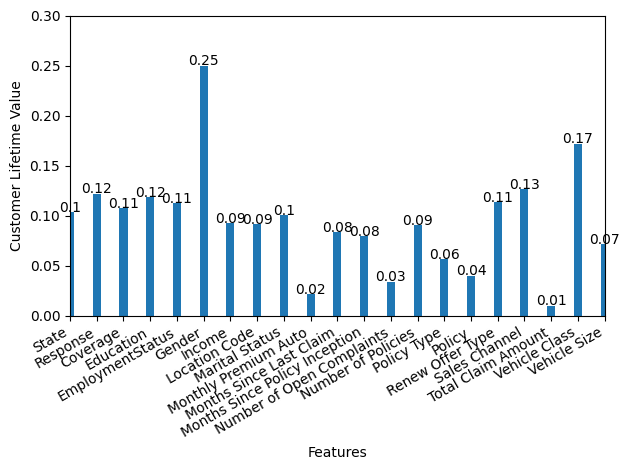
# Exploratory Data Analysis

1. *Data Overview*

Here we have both categorical and numerical variables in our data. The numerical variables are: Customer Lifetime Value, Income, Monthly Premium Auto, Months Since Last Claim, Months Since Policy Inception, Number of Open Complaints, Number of Policies, and Total Claim Amount. The categorical variables are: Customer, State, Response, Coverage, Effective to Date, EmploymentStatus, Gender, Location Code, Marital Status, Policy, Policy Type, Renew Offer Type, Sales Channel, Vehicle Class, and Vehicle Size. After finding which variables are categorical and which are numerical, we could then look at the relationship between these variables and our target Customer Lifetime Value. First, we focused on calculating the correlation between the numeric variables and the target. From this we found that the three top correlated variables were "Monthly Premium Auto", "Number of Policies", and "Coverage". The highest of these was "Monthly Premium Auto" at approximately 0.37. Both barplots and boxplots were used to find the relationship between the target and the categorical variables. With the barplots we were able to see the data in its entirety while with the boxplots we could see the data and its outliers. Both the boxplots and the barplots were used before the outliers were removed in order to see the data in its entirety.

The two different plots led us to choose different variables from the data. From the boxplots, we found change in relation to the target variable in "EmploymentStatus", "Renew Offer Type", "Vehicle Class", and "Policy". The analysis done with the barplots gave us the variables "Marital Status", "Renew Offer Type", "Vehicle Class", and "Vehicle Size". From these observations we created two different variable selections. The first contained the variables "EmploymentStatus", "Renew Offer Type", "Vehicle Class", "Policy", "Monthly Premium Auto", and "Coverage". The second selection of variables were "Coverage", "Marital Status", "Monthly Premium Auto", "Months Since Last Claim", "Months Since Policy Inception", "Number of Open Complaints", "Number of Policies", "Renew Offer Type", "Total Claim Amount", "Vehicle Class", "Vehicle Size". Figures of the boxplot EDA and the barplot EDA are found in the appendi

1. Feature Selection

Considering the challenges related to the dimensionality of our dataset, along with the fact that our initial models were not achieving reasonable results, we continued to explore other tools that could offer better insights either by eliminating or transforming existing variables. From the exploratory analysis performed during EDA, since most of the variables did not have strong correlations among one another, we believed that a more robust feature selection would allow us to build stronger predictive algorithms. Initially, we conducted Lasso and Ridge regression tests to verify the best alpha, and with that, also calculate the coefficient of importance of our variables. Because of the insufficient results, we decided to implement an automated algorithm. For this approach, we decided to build a wrapper method, since it would offer the possibility to simultaneously evaluate diferent criterias of our variables. We used both Forward selection and Backward selection as well. By starting with Forward selection we tried to guarantee that the process would consider one feature each time, depending on the level of significance, and then repeat the process for all features progressively. In the same process, we checked the results for Backward Selection, which follows an opposite approach, but surprisingly, we observed small differences when applied to our datasets. For this we used the Sequential Feature Selector builtin tool in Python. Then, given the lack of considerable improvements from feature selection techniques, mostly because this *Figure 1: Important Features of the dataset*

dataset is non-normally distributed, we decided to pursue other techniques to improve the performance of our models, which will be explained in the next sections.

# Principal Component Analysis (pca)

With the goal of enhancing our predictive power, we decided to explore PCA. The main goal was to reduce the noise, aid the visualization of our clusters, and proceed with feature engineering. This last technique is especially important to us because it focused on decreasing the number of features, even if it could represent the increase in the dimension of others. For the first try, we used the PCA library from Sklearn. In our dataset, just a few variables are easily identified for having a high percentage of explained variance. This might be the reason why the number of principal components required by the algorithm was unnecessarily too high. After trying this approach for a few days, we noticed that our models trained with the components could have been overfitting. On the other hand, if we decreased the number of components, the models would yield extremely low scores. We then decided to explore other libraries with the same goals, which required us to select the smallest amount of components as possible and maintain a good amount of variance explanation.

# Model Analysis

Next we wanted to test how the different versions of the exploratory data analysis and the different clusters worked with different models. For this we tested Lasso regression, Ridge regression with elastic net, Generalized Linear model, Decision Tree, Random Forest, and Gradient Boosted Trees. In each section we experimented with the different versions of the exploratory data analysis and the different clusters we created. Each of these models were trained using twenty percent of the data dedicated to training and the remaining eighty percent for testing the model. Log normalization was also implemented for all models except for the generalized linear model.

Before testing the models we wanted to see how the different versions of the exploratory data analysis affected the baseline model. For this test we used all of the models on all of the different versions of the EDA. On average, the models that had no EDA applied to them got lower RMSE than those that had variables dropped. The EDA from the barplots on average performed better than those with the boxplot EDA variables.

## Lasso Regression

The first model that was tested with the different combinations of the data was Lasso regression. We used a model that had all of the variables in order to establish a baseline model. We then tested each of the clusters with their own separate model, using the variables from the barplot EDA, the boxplot EDA, and the baseline model. The resulting average root mean square errors for each of the different types of EDA were 4,779.21 for no EDA, 4,792.01 for the barplot EDA, and 4,958.82 for the boxplot EDA. As for the different clusters tested with this model, Policies and Vehicle did the best at 4,691.33 and 4,767.45 respectively.

*B. Ridge Regression with Elastic Net*

The next step after using the Lasso model was using the Elastic Net model. From here we tested it with the same combinations of our data from Lasso and compared the different models using root mean square error. Here the resulting average RMSE is 4,99.47 for all of the variables, 5,007.59 for the bar plot EDA variables, and 5,147.26 for the boxplot EDA variables. For the clusters, Vehicle and Policies attained the lowest scores with 4,965.17 RMSE belonging to Vehicle and 4,982.06 RMSE for Policies. Overall, the RMSE of the Elastic Net was higher than that of the Lasso model, showing that this model may not be a good fit for our data.

*C. Generalized Linear Model (GLM)*

The next model we used was the Generalized Linear Model. This model used a log link function since the target distribution was inverse Gaussian. Once again, the same combinations of the EDA were used to test the model. Here the model built with the variables from the boxplot EDA got the lowest RMSE at 4,535.18. The next lowest model was built using the variables from the barplot EDA with an RMSE of 4,681,84 and the GLM built using all of the variables got an RMSE of 4,918.72. The clusters that did the best with this model were the Vehicle cluster at 4,595.07 RMSE and the Area cluster at 4561.75 RMSE. The pattern of the RMSE for the EDA was different from the Lasso and Elastic Net regression where the variables from the boxplot analysis got the lowest RMSE.

*D. Decision Tree*

After using linear models on our data, we decided to also test the data with non-linear models to see if they could better fit the data. A grid search algorithm was used for each decision tree so that a model could be created that better fit the data. This controlled two things: the maximum depth the tree could grow and the maximum number of leaf nodes the tree could have. The maximum depth spanned a range from two to seven and the maximum number of leaf nodes spanned from four to nine. The Decision Tree with all of the variables performed better than the two where EDA was used to reduce the variables with an average test RMSE of 3175.89 and an average training RMSE of 3086.8. As for the clusters, Vehicle and Area both got the lowest average test RMSE of 3,633.95 and 3,715.69 respectively. The average training RMSEs for Vehicle and Area were close to this value at 3,643.45 and 3,621.28.

*E. Random Forest*

Next, we fitted the different data combinations to the Random Forest model. This model also had grid search applied to it in order to find the best parameters. In this model the number of trees was fixed to one-hundred and the maximum amount of features to use was calculated applying the logarithmic function. The maximum depth of the trees ranged from six to nine and the maximum number of leaf nodes ranged from four to nine. The lowest RMSE for the Random Forest was achieved using the bar plot EDA variables at a value of 4,099.07 with an average training RMSE of 3,922.99. As for the clusters, the resulting root mean square errors were very similar, except for Vehicle, which achieved an RMSE of 4,295.76. The average training RMSE for this cluster was 4,347.19.

*F. Gradient Boosting Regressor*

Lastly, the Gradient Boosted Tree Regression model was used to fit the data. This also utilized grid search, however, the number of estimators was set to one-hundred, the learning rate was set to 0.05, and the function to find that maximum number of features was a logarithmic function. The estimators that varied were the maximum depth of the trees ranging from six to nine and the maximum number of leaf nodes ranging from four to nine. Using the variables from the barplot EDA got the lowest average testing RMSE of 4,289.11 and a training RMSE of 4,277.5. The cluster that got the lowest average testing RMSE was Vehicle at 4,436.49 with a training RMSE of 4,489.55.

# Interpretation Of Results

The results are interpreted as under and to determine which model is better, let's first understand Root Mean Squared Error (RMSE), the evaluation metric used. RMSE measures the average magnitude of the errors between predicted and actual values, and lower RMSE values indicate better model performance.

| **Clusters** | **Lasso (No EDA)** | **Lasso (Bar)** | **Lasso (Box)** | **E Net (No EDA)** | **E Net (Bar)** | **E Net (Box)** | **GLM (No EDA)** | **GLM (Bar)** | **GLM (Box)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No Cluster | 4932.17 | 5058.43 | 4935.92 | 5113.15 | 5227.86 | 5082.39 | 4742.15 | 4629.91 | 4605.16 |
| Vehicle | 4721.52 | 4552.24 | 5028.59 | 4917.36 | 4764.73 | 5213.41 | 4661.03 | 4736.34 | 4390.44 |
| Cost | 4846.78 | 5055.06 | 4899.12 | 5059.82 | 5225.88 | 5065.31 | 4707.48 | 4558.98 | 4642.41 |
| Area | 4729.44 | 4710.30 | 5106.68 | 4907.80 | 4935.76 | 5311.91 | 4580.64 | 4494.34 | 4610.28 |
| Policies | 4666.14 | 4584.03 | 4823.81 | 4999.23 | 4883.70 | 5063.26 | 5902.28 | 4989.66 | 4427.59 |
| AVERAGE | 4779.21 | 4792.01 | 4958.82 | 4999.47 | 5007.59 | 5147.26 | 4918.72 | 4681.85 | 4535.18 |

*Table 1 Testing RMSE For Regression Models*

| Clusters | DT (No EDA) | DT (Bar) | DT (Box) | RT (No EDA) | RT (Bar) | RT (Box) | BT (No EDA) | BT (Bar) | BT (Box) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No Cluster | 3090.42 | 3282.75 | 4864.04 | 4441.36 | 4257.38 | 4898.13 | 4508.90 | 4456.77 | 4928.04 |
| Vehicle | 2990.85 | 2952.30 | 4958.69 | 4233.52 | 3673.90 | 4979.86 | 4310.21 | 3960.31 | 5038.96 |
| Cost | 3144.94 | 3215.26 | 4879.37 | 4243.16 | 4126.75 | 4901.59 | 4405.57 | 4360.85 | 4926.82 |
| Area | 3013.32 | 3087.74 | 5046.02 | 4169.68 | 4089.91 | 5098.21 | 4286.40 | 4219.01 | 5141.77 |
| Policies | 3639.90 | 4162.35 | 4551.46 | 4382.64 | 4347.42 | 4609.38 | 4409.78 | 4448.60 | 4688.71 |
| Average | 3175.89 | 3340.08 | 4859.92 | 4294.07 | 4099.07 | 4897.43 | 4384.17 | 4289.11 | 4944.86 |

*Table 2 Testing RMSE for Tree-Based Models*

| Clusters | Lasso (No EDA) | Lasso (Bar) | Lasso (Box) | E Net (No EDA) | E Net (Bar) | E Net (Box) | GLM (No EDA) | GLM (Bar) | GLM (Box) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No Cluster | 4779.24 | 4745.35 | 4793.47 | 4980.05 | 4952.11 | 4983.31 | 4596.60 | 4628.59 | 4653.98 |
| Vehicle | 4829.88 | 4866.81 | 4777.82 | 5025.08 | 5053.45 | 4955.47 | 4616.51 | 4608.02 | 4712.48 |
| Cost | 4789.82 | 4750.21 | 4810.69 | 4986.26 | 4951.83 | 4992.71 | 4608.56 | 4643.44 | 4649.55 |
| Area | 4829.18 | 4831.96 | 4757.02 | 5036.91 | 5014.42 | 4932.51 | 4630.88 | 4658.41 | 4654.40 |
| Policies | 4671.42 | 4749.78 | 4785.45 | 5007.85 | 5035.40 | 5002.71 | 5852.50 | 4993.54 | 4630.37 |
| Average | 4779.91 | 4788.82 | 4784.89 | 5007.23 | 5001.44 | 4973.34 | 4861.01 | 4706.40 | 4660.15 |

*Table 3 Training RMSE for Regression Models*

| Clusters | DT (No EDA) | DT (Bar) | DT (Box) | RT (No EDA) | RT (Bar) | RT (Box) | BT (No EDA) | BT (Bar) | BT (Box) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No Cluster | 3086.80 | 3034.01 | 4739.89 | 4297.84 | 3922.99 | 4773.70 | 4368.55 | 4138.26 | 4809.88 |
| Vehicle | 3104.54 | 3108.25 | 4717.57 | 4348.89 | 3958.94 | 4733.74 | 4428.80 | 4253.83 | 4786.01 |
| Cost | 3072.81 | 3058.56 | 4747.06 | 4146.99 | 3835.57 | 4792.50 | 4313.98 | 4071.98 | 4825.04 |
| Area | 3091.05 | 3076.84 | 4695.94 | 4276.64 | 4186.96 | 4747.28 | 4400.10 | 4309.09 | 4779.83 |
| Policies | 3772.44 | 4320.14 | 4569.62 | 4441.34 | 4347.42 | 4600.93 | 4460.69 | 4614.35 | 4659.55 |
| Average | 3225.53 | 3319.56 | 4694.02 | 4302.34 | 4050.38 | 4729.63 | 4394.42 | 4277.50 | 4772.06 |

*Table 4 Training RMSE for Tree-Based Models*

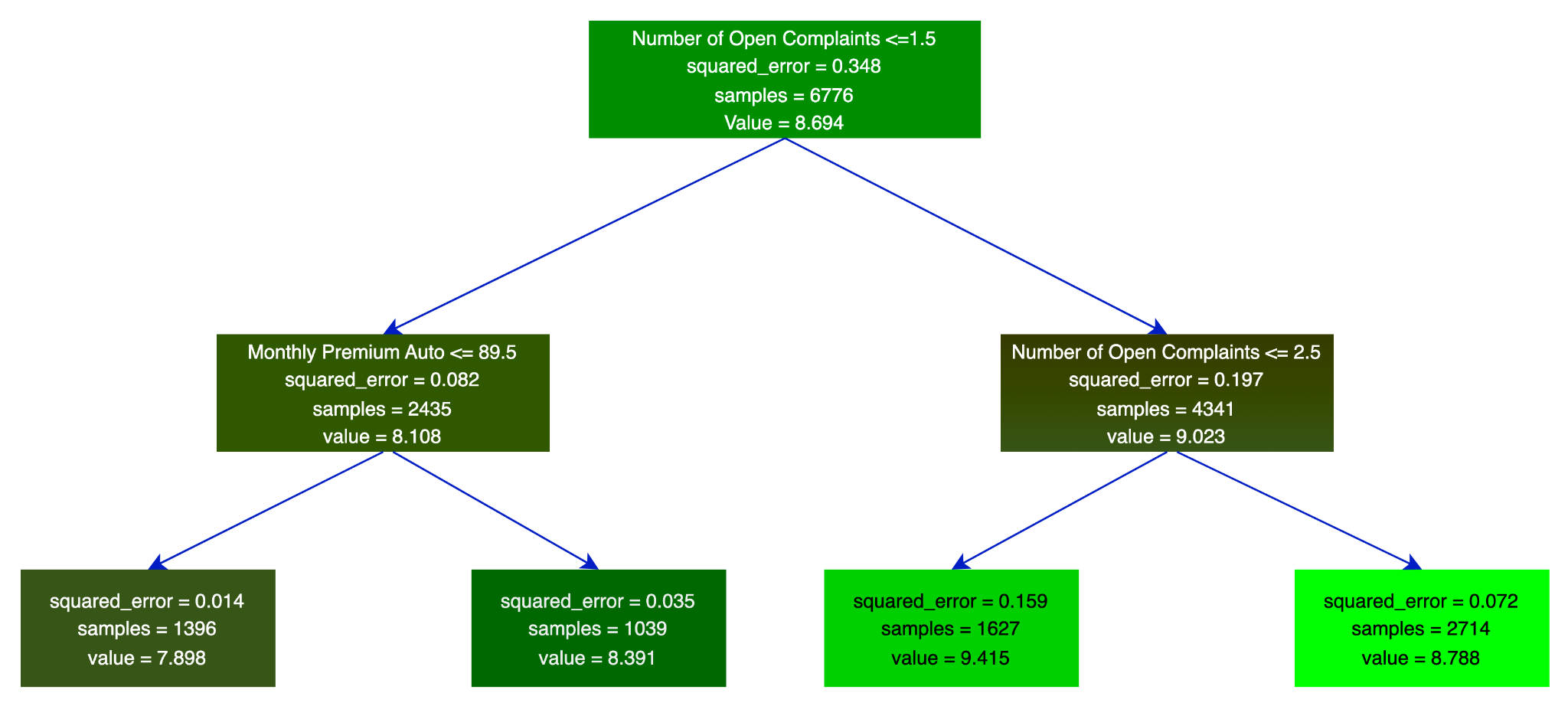
Considering Table 1 and Table 3 for training and testing RMSE values on regression models ,the Testing RMSE in regression models, the model with the lowest average testing RMSE is the "GLM (Box)" model, with an average testing RMSE of approximately 4535.18. This model seems to perform the best on the testing dataset for predicting customer lifetime value across different clusters.For Training RMSE the model with the lowest average training RMSE is the "GLM (No EDA)" model, with an average training RMSE of approximately 4596.60. This model appears to have the best performance on the training dataset. The "GLM (Box)" model has the lowest testing RMSE, indicating good performance on unseen data, while the "GLM (No EDA)" model has the lowest training RMSE, indicating a good fit to the training data. The "GLM (No EDA)" model also has a relatively low testing RMSE, suggesting that it generalizes well to unseen data despite its slightly better performance on the training data. Considering both testing and training RMSE values, the "GLM (Box)" model seems to be the most appropriate choice for predicting customer lifetime value. It strikes a good balance between generalization on unseen data and fit to the training data.

Considering the results interpreted from Table 2 and Table 4. The given results represents the Root Mean Squared Error (RMSE) values for different models (Decision Tree - DT, Random Forest - RT, and Boosted Tree - BT) with and without Exploratory Data Analysis (EDA), along with their corresponding RMSE values for specific variables (No Cluster, Vehicle, Cost, Area, and Policies).

Overall Observations:

* Decision Tree (DT) tends to have the lowest RMSE values in most scenarios, especially when EDA is performed (Barplot and Boxplot).
* Among Decision Tree (DT), Random Forest (RT), and Boosted Tree (BT) models, Decision Tree (DT) generally performs better in most scenarios, except for the "No Cluster" variable where Random Forest (RT) has the lowest average RMSE value.
* EDA (Barplot and Boxplot) tends to lead to higher RMSE values for all the models, indicating potential overfitting or inclusion of irrelevant features in the EDA process.
* The performance of the models varies across different variables, with "No Cluster" having the lowest average RMSE value, indicating that the models perform better on this variable compared to the others.
* Decision Tree (DT) tends to have the lowest average RMSE values in most scenarios, especially when EDA is performed (Barplot and Boxplot).
* Random Forest (RT) and Boosted Tree (BT) have higher average RMSE values compared to Decision Tree (DT) in most scenarios.
* EDA (Barplot and Boxplot) still tends to lead to higher average RMSE values for most models, indicating potential overfitting or inclusion of irrelevant features in the EDA process.

Since the tree models did better than the linear models and the clusters, Vehicle and Area, did well for the tree models, we decided to make one final model that contained both the Vehicle and Area clusters. From this we tested the trees using all the variables and the variables from the barplot analysis since those two worked well for trees. The model with the lowest RMSE was a Decision Tree using the variables Coverage, Marital Status, Monthly Premium Auto, Months Since Last Claim, Months Since Policy Inception, Number of Open Complaints, Number of Policies, Renew Offer Type, Total Claim Amount, and the clusters Area and Vehicle. The resulting tree from this was very small and only used Number of Policies and Monthly Premium Auto to predict.



*Figure 2 Final Model: Decision Tree*

# Conclusion

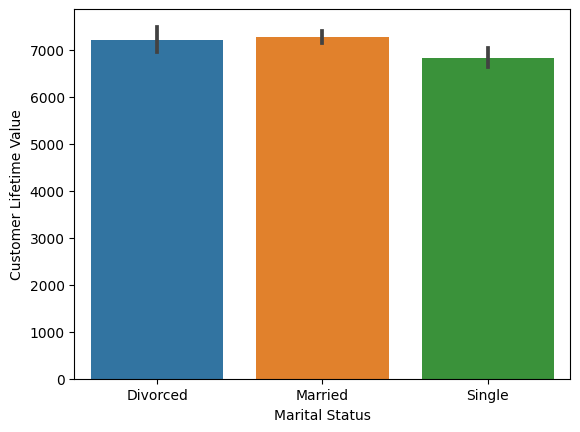
In this paper we aimed to predict the Customer Lifetime Value using the data from the IBM Watson Marketing Customer Value dataset. We did an exploratory data analysis on the data and reduced the dimensionality through feature selection and clustering. Based on that we built various models, namely GLM, K-Means, Decision Tree, Random Forest, Boosted Tree, Lasso, and Elastic Net. The primary evaluation metric used was Root Mean Squared Error (RMSE), with lower RMSE values indicating better model performance. The findings from the evaluation of different models and their corresponding RMSE values are as follows:

* Decision Tree (DT) demonstrated the best performance among all models, consistently achieving the lowest RMSE value of 3090.423445.
* Random Forest (RT) and Boosted Tree (BT) showed reasonable performance, with RMSE values of 4441.355364 and 4508.899339, respectively.
* GLM performed less accurately than Decision Tree, Random Forest, and Boosted Tree, with an RMSE of 4742.153577.
* LASSO and Elastic Net had higher RMSE values compared to Decision Tree but were relatively close to GLM, with RMSE values of 4932.169408 and 5113.145916, respectively.

Further analysis was performed using exploratory data analysis (EDA), including barplots and boxplots. The results from EDA indicated that Decision Tree (DT) remained the model with consistently low RMSE values in both barplots and boxplots. Random Forest (RT) showed good performance in barplots but had a relatively higher RMSE in boxplots. Boosted Tree (BT) performed similarly to Random Forest but still trailed behind Decision Tree. Additionally, models with EDA tended to have slightly higher average RMSE values, suggesting potential overfitting or the inclusion of irrelevant features during EDA. The performance of the models also varied across different variables. As for the clusters, the Vehicle cluster got the lowest average RMSE value, showing that the K-means was effective in improving model performance.

In conclusion, based on the analysis of the RMSE values and the EDA results, the Decision Tree model (DT) is the best performer among the models considered for predicting Customer Lifetime Value in the IBM Watson Marketing Customer Value Data project. The project further highlights the importance of proper data cleaning, feature engineering, and model selection in achieving accurate predictions of customer lifetime value. Moreover, it sheds light on the potential impact of exploratory data analysis on model performance, emphasizing the need for careful consideration of its application to avoid potential overfitting or the inclusion of irrelevant features.

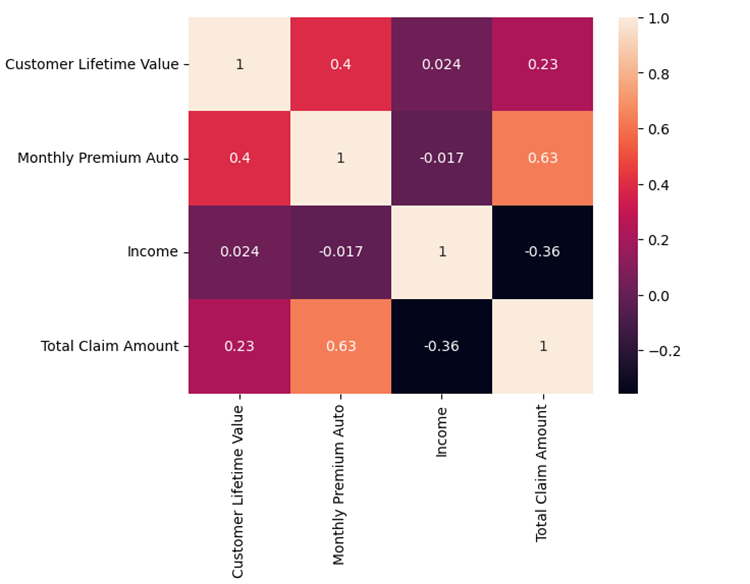
# Executive Summary

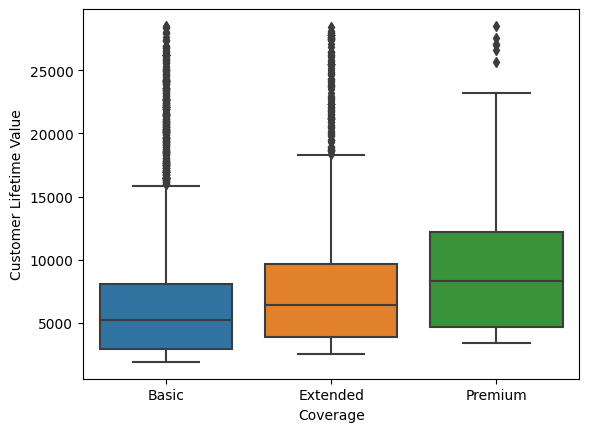
In this paper, our team detailed our work on predicting the Customer Lifetime Value of a customer. First, we did an exploratory data analysis in order to obtain a better understanding of the data and see what we could use to predict the target variable, Customer Lifetime Value. From this, we came up with two different understandings based on whether we use the outliers in the data or not when looking at the relations between the data and our target. The first data analysis which we named ‘the boxplot analysis’ after the graphs used to look at the variables, excluded the outliers. The other analysis we called the barplot analysis since the barplots also included the outliers when comparing the relationships between the data and the target. Next, we tried to use Principal Component Analysis in an attempt to improve the models we used later. This did not lead to better results so we decided to try other methods. We then tried K-means clustering in order to group like variables together. From this, we created new variables from the old ones, such as Vehicle that described the type of vehicle the customer drove, Area that gives a more detailed location of where the customer lives, Cost which tells us how much of a risk the customer is by looking at how much they have claimed in total and the how much time has passed since the last claim, and finally Policies that combines all of the different features relating to the type of policy the customer has with the number of policies the customer owns. After this, we tried a bunch of different models with the different combinations of the analysis and the clusters. The models we used were Lasso regression, Elastic Net regression, Generalized Linear Model, Decision Tree regression, Random Forest regression, and Gradient Boosted Trees. We found that out of all the possible combinations that the Decision Tree regression scored better than all of the models. This model used the variables that we obtained from the barplot analysis which were Coverage, Marital Status, Monthly Premium Auto, Months Since Last Claim, Months Since Policy Inception, Number of Open Complaints, Number of Policies, 'Renew Offer Type, Total Claim Amount, Vehicle Class, and Vehicle Size. From this we found that the model considered the Number of Open Complaints and the Monthly Premium Auto to be the most important features, with the Number of Open Complaints being more important than the Monthly Premium Auto.

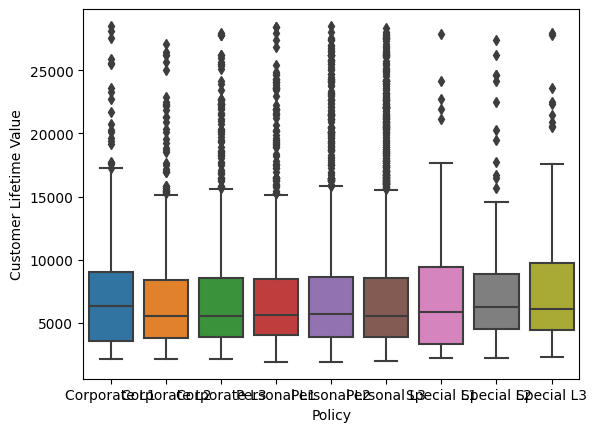
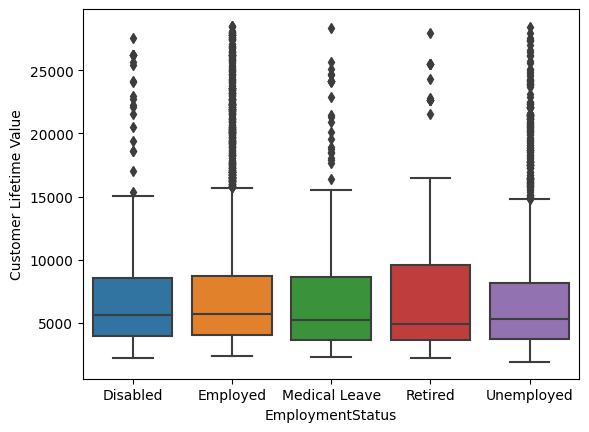
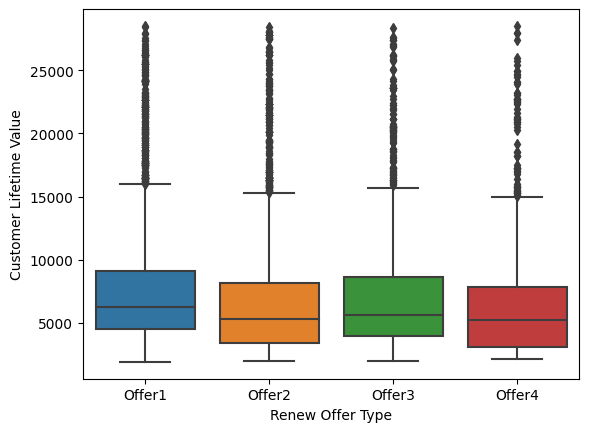
# Appendix

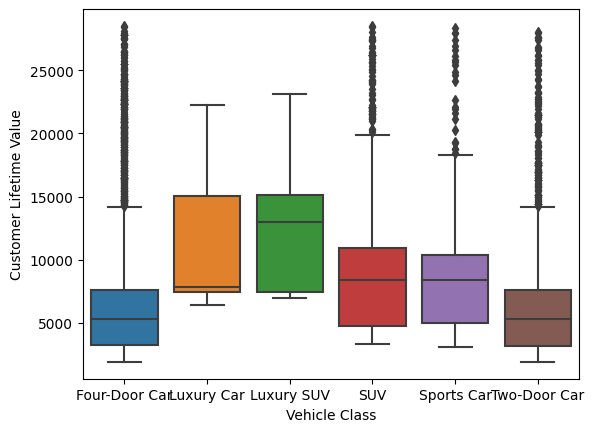
[1]h[ttps://www.kaggle.com/datasets/pankajjsh06/ibm-watson-marketing-customer-value-data](https://www.kaggle.com/datasets/pankajjsh06/ibm-watson-marketing-customer-value-data)

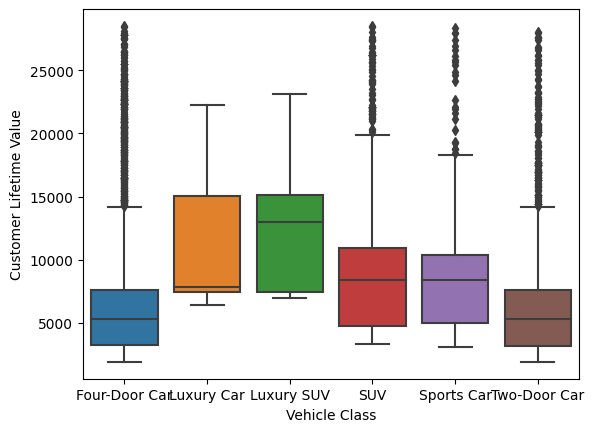
[2] Figure 3: Heatmap: correlation of Exploration Data Analysis

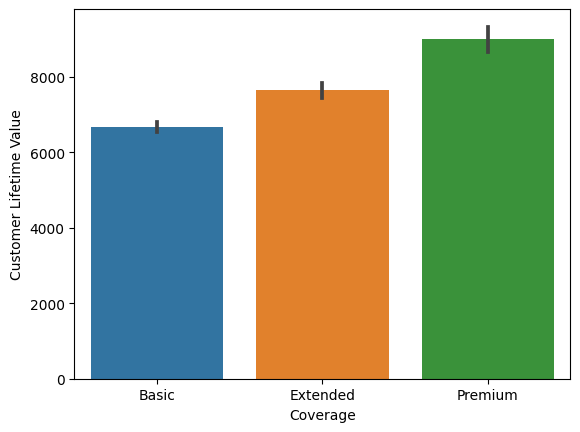
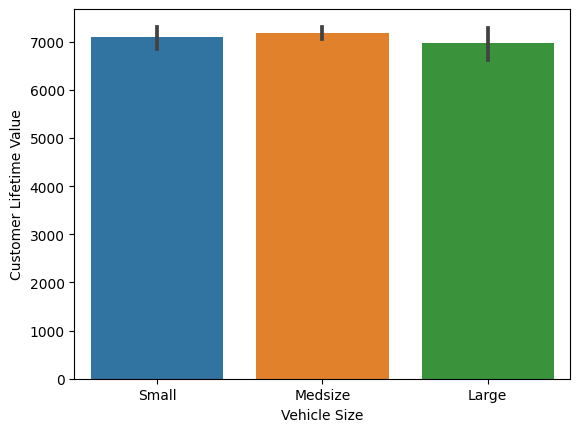


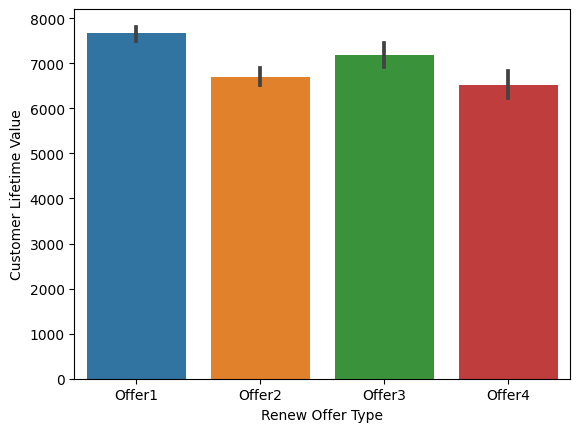
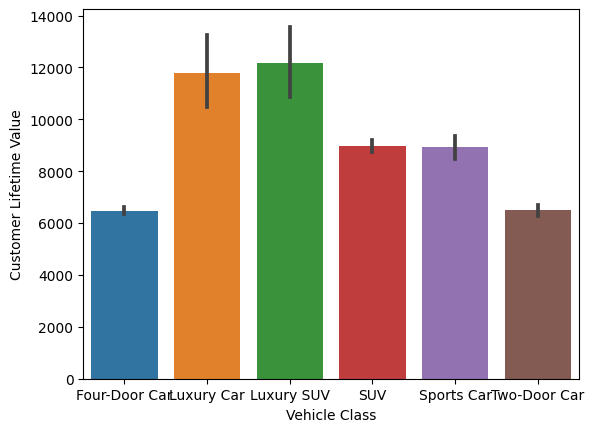
[3] Figure 4: Boxplots: Exploration Data Analysis



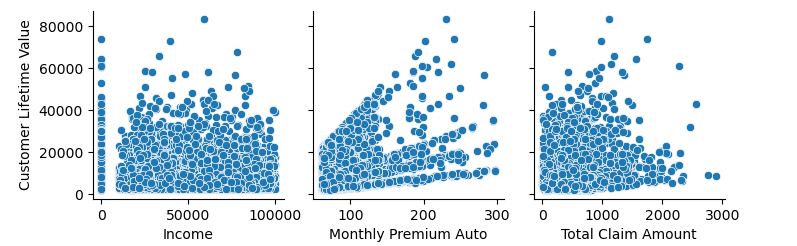


[4] Figure 5: Barplots: Exploration Data Analysis 

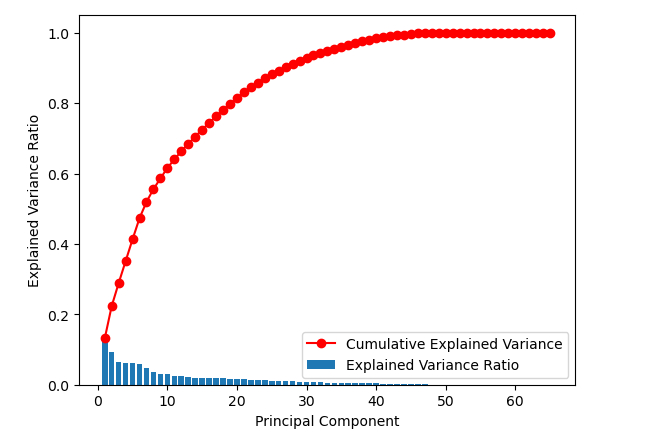




[5] Figure 6: Scatterplot: Exploration Data Analysis



[6] Figure 7: Variance Explained Ratio: Principal Component Analysis

[7] Figure 8: Biplot:Principal Component Analysis

