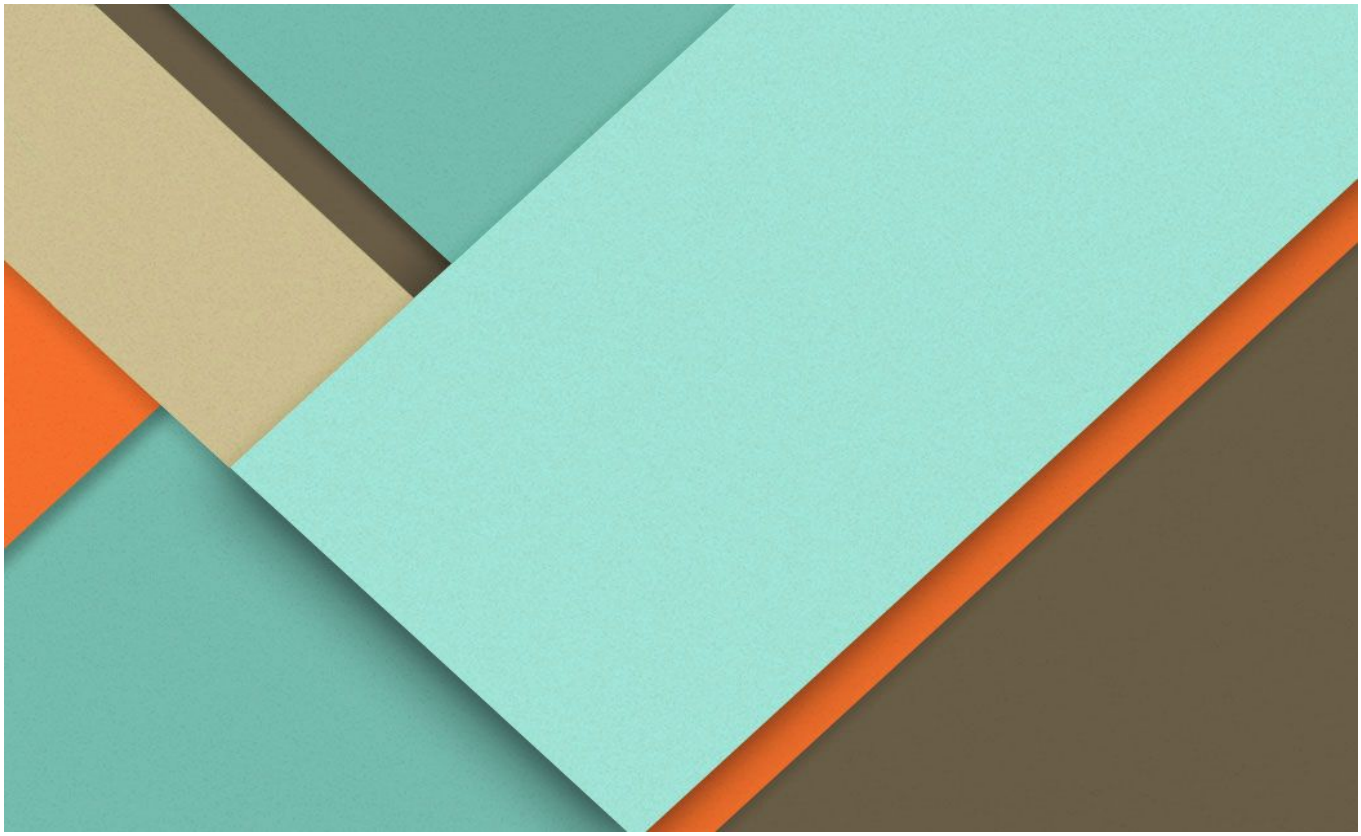


TEAM-**Abraca-data**

INDEX

PROJECT OVERVIEW

WHAT IS CUSTOMER LIFETIME VALUE ?

The lifetime value of a customer, or customer lifetime value (CLV), represents the total amount of money a customer is expected to spend in your business, or on your products, during their lifetime. This is an important figure to know because it helps you make decisions about how much money to invest in acquiring new customers and retaining existing ones.

In the big picture, CLV is a gauge of the profit associated with a particular customer relationship, which should guide how much you are willing to invest to maintain that relationship.

WHAT IS ITS IMPORTANCE ?

Customer Lifetime Value tells you how well you're resonating with your audience, how much your customers like your products or services, and what you're doing right — as well as how you can improve. Customer lifetime value is important because, the higher the number, the greater the profits. You'll
always have to spend money to acquire new customers and to retain existing ones.

PROBLEM STATEMENT

In this project, we are given the geodemographic and other relevant data of an Auto Insurance Company that may impact its Customer Lifetime Value. We have to predict the Customer Lifetime Value for this company. Customer Lifetime Value is the total revenue the client will derive in its entire relationship with a customer. Because we don't know how long each customer relationship will be, we have to make a good estimate and state Customer Lifetime Value as a periodic value - that is, we usually say "that this customer's 12 month(or 24 months, etc) Customer Lifetime Value is $x.

The client also wants to know the types of customers that would generally give us more revenue.

DATA DESCRIPTION

There are **24** total columns in the dataset given.
Here is the description of each column:-

- **Customer:** Unique Customer ID for each customer.
- **State:** The state in which the customer lives.
- **Customer Life** : Time Value: The total revenue the client will derive the entire relationship with a customer.
- **Response:** Response of the Customer.
- **Coverage:** The amount of risk or liability that is covered for an individual or entity by way of insurance services.
- **Education:** Highest educational qualification of a customer.
- **Effective To Date:** The date to which the insurance is effective.
- **Employment Status:** Employment Status of the customer.

- **Gender:** Gender of the customer.
- **Income:** Income of the customer.
- **Location Code:** The region of the customer i.e. urban, suburban or rural.
- **Marital Status:** Marital Status of a person, that is, whether a person is married, single, divorced, etc.
- **Monthly Premium Auto:** Monthly premium of the insurance.
- **Months Since Last Claim:** Number of months it has been since a customer last claimed their insurance.
- **Months Since Policy Inception:** Number of months since the policy was taken by the customer.
- **Number of Open Complaints:** Number of complaints filed by the customers for claiming insurance.
- **Number of Policies**: Total number of policies undertaken by the customer.
- **Policy Type**: The type of policies taken i.e. corporate, personal, or special.
- **Policy**: Various schemes under which Customers enroll.
- **Renew Offer Type:** Type of renew offer.
- **Sales Channel:** A method of distribution used by a business to sell its products, that is agent, call center, etc.
- **Total Claim Amount:** Total amount of insurance money claimed by the customer.
- **Vehicle Class:** Class of the vehicle i.e. two-door, four-door, SUV, Luxury SUV, Sports or Luxury.
- **Vehicle Size:** The size of each vehicle of the customer insured by the company.

EVALUATION METRIC

ROOT MEAN SQUARE ERROR

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results. The formula for RMSE is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Here we calculate the predicted value for each test data set. Then we subtract the actual value of the objective from the predicted value and square it. Then we divide by the total number of training data sets. Then evaluate the square root of the result which gives us the RMSE.

MEAN ABSOLUTE PERCENTAGE ERROR

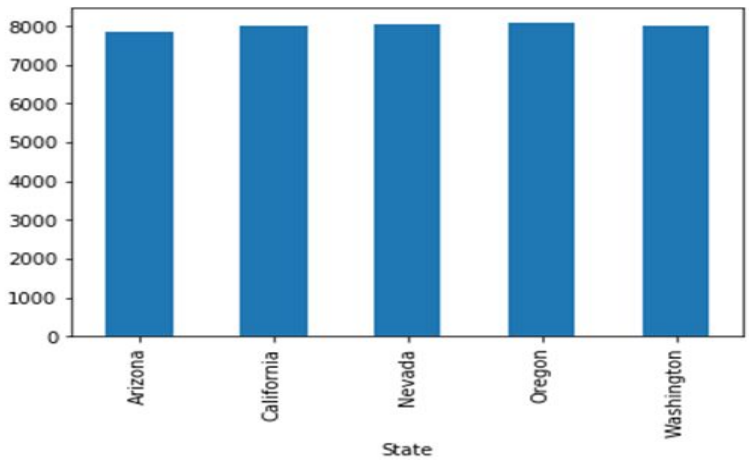
The *mean absolute percentage error* (**MAPE**) is a statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values. Where $\mathbf{A_t}$ is the *actual value* and $\mathbf{F_t}$ is the *forecast value*, this is given by:

$$\mathrm{M} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

ANALYSIS

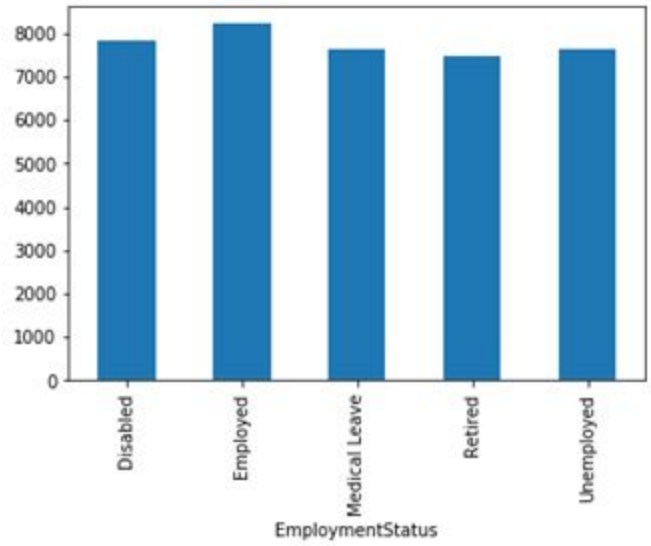
DATA EXPLORATION

The dataset given is a matrix of **9134x24** dimension.From the dataset,we have identified and set our target variable **y** .i.e. Customer Lifetime Value.
Rest of the features were assigned to a variable **x**. In order to realise the relationship between the target variables and various features, we plotted some graphs between them, some of which are shown below. Analysing the various data, we will apply feature engineering and clean the data accordingly.



X-Axis: State **Y-Axis:** Mean CLV

There is not much difference between the mean CLV among the customers in the different states.



X-Axis: Education **Y-Axis:** Mean CLV

From the Bar Graph, we see that the company would benefit on targeting the employed section of its customers as they provide the maximum CLV amon the other categories.

FEATURE ENGINEERING

Feature engineering is about creating new input features from your existing ones.In general, Feature Engineering is a process of addition and Data Cleaning is a process of subtraction.

After plotting the graphs between various features in **x** and the target variable **y** (shown in data exploration), we realised that some of the features were not as useful as others in the prediction of our data.Thus, we removed some of the features from **x** .i.e. Gender and State.

Our dataset has now become a matrix of **9134x22** dimensions.Doing this has made our computations faster than before.

Next,we noticed that the column **Customer** has all unique values and has a string attached to a bunch of numbers. So we differentiated the string part into a separate column, **New Parameter**.We have made another column **Average_ID**

where we have computed the average CLV for each string part in the column, **New Parameter**.

We have noticed that doing this has decreased our RMSE value significantly and has also resulted in faster computations.

We have also calculated the Present Value of each customer and put them into a separate column **Present Value.** This gives us the worth of each customer to the company and helps the company focus on which customers they should focus more. We have calculated it by the following formula

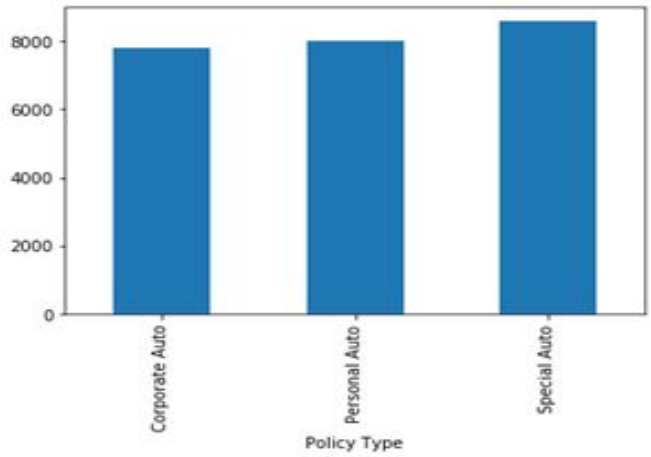
*Present Value=Monthly Premium Amount*Months Since Policy Inception - Total Amount Claim*

We also did a bit of Hyper-Parameter Tuning in Random Forest which is elaborated more in the Model Selection section.

EXPLORATORY ANALYSIS

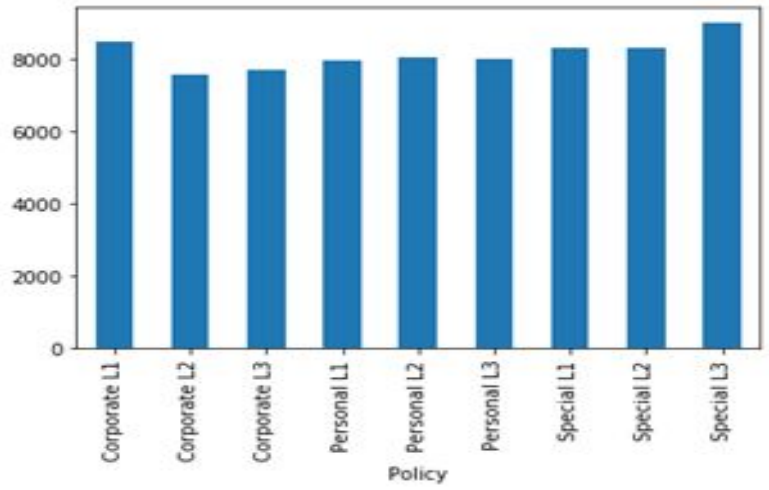
X-Axis: Coverage **Y-Axis:** Mean CLV

The Customers having premium coverage provide Maximum mean CLV.



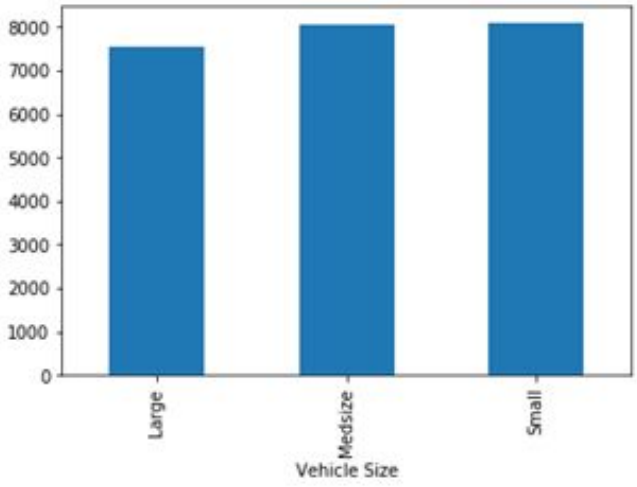
X-Axis: Policy Type **Y-Axis:** Mean CLV

Special Auto Policy Type Customers give the Maximum mean among the policies.



X-Axis: Policy Type **Y-Axis:** Mean CLV

Special L3 customers have the highest Average CLV among the various categories of policies.
Thus, Special Auto + Special L3 customers should provide the highest CLV among the Customers. Thus, the customers who have enrolled within these two categories would be an invaluable asset for the company.



X-Axis: Vehicle Size **Y-Axis:** Mean CLV

The Large Size vehicles doesn't procure as much CLV as the other vehicle sizes.
These are the analysis that we realised upon Exploratory Analysis.

ALGORITHMS & TECHNIQUES

DATA PREPROCESSING

Data Preprocessing is a technique that is used to convert the raw data into a clean data set.

➢ **Splitting dataset into train-set and test-set**
The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

- **xTrain** is the training data set.
- **yTrain** is the set of labels to all the data in **xTrain**.
- **xTest** is the selected data set.
- **yTest** is the set of labels predicted to all data in **xTest.**

Syntax:

from sklearn.model_selection **import** train_test_split X_train X_tr, X_test, y_train, y_test = **train_test_split**(x, y, test_size=0.2, random_state=0)

➢ **Label Encoding**

Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form.

Syntax

```
# Import label encoder
from sklearn.preprocessing import LabelEncoder

# label_encoder object knows how to understand word  labels.
```

label_encoder =LabelEncoder()

 # Encode labels in the column you need to (for example,'species')
df['species']= label_encoder.fit_transform(df['species'])
df['species'].unique()

➢ **One-Hot Encoding**

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.Just label encoding is not sufficient to provide to the model for training because as the number of unique entries increases, the categorical values also proportionally increases.The higher the categorical value it assumes , better the category for the model.This could lead to inaccuracies and wrong results.

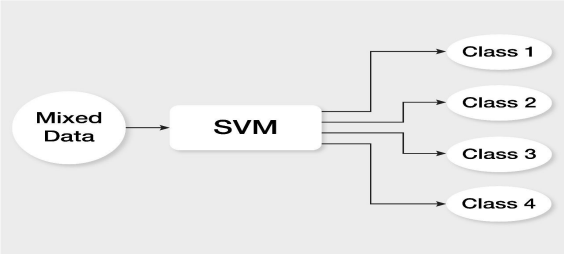
Syntax:

onehot_encoder = OneHotEncoder(sparse=False)
integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
onehot_encoded = onehot_encoder.fit_transform(integer_encoded)

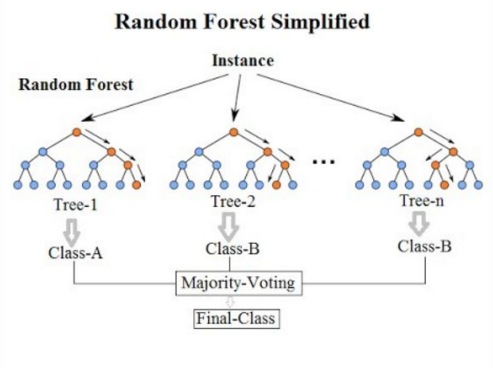
MODEL SELECTION

The Model **Random Forest Regressor** was best fitted for the prediction of the "Customer Life-time Value" . For experimental purposes , we used all the models described below :

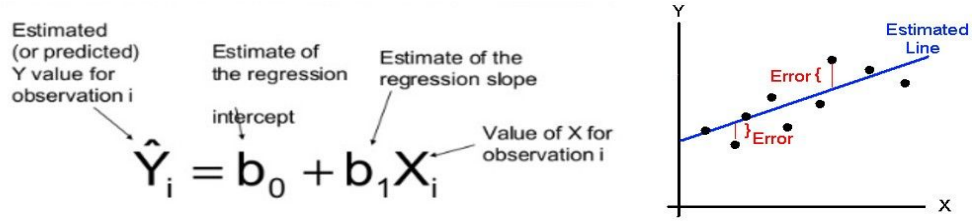
- **Support Vector Machine(SVM)**:"Support Vector Machine" (**SVM**) is a supervised machine learning algorithm which can be used for both classification or regression challenges. The idea of **SVM** is simple: The algorithm creates a line (or a hyperplane) which separates the data into classes.



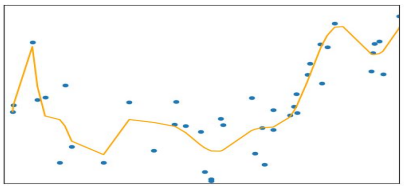
- **Random Forest:** Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.



- **Linear Regression:** Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.



- **Polynomial Regression:** Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an *nth* degree polynomial. These are basically used to define or describe nonlinear phenomenon.



PRECAUTIONS

While applying our model, we must be wary of overfitting our data. This would lead to large inaccuracies in our prediction. **Overfitting** is basically a modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study.
In reality, the data often studied has some degree of error or random noise within it. Thus, attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power.

MODEL PERFORMANCE

MODEL	RMSE	MAPE
Linear Regression	3746.93	11.8%
Polynomial Regression	5760	44.12%
Support Vector Machine	6811.54	55.63%
Random Forest	3523.17	9.67%

Table of Models with respective RMSE and MAPE value

For obtaining the said RMSE and MAPE in **Random Forest**,we have also done a bit of Hyper-Parameter Tuning and found out that the parameters for which it gave the best RMSE were n_estimators=100, random_state=0,min_samples_split=20.

RESULT

Clearly,according to the table drawn in the section **Model Selection,** the model which has performed best on the given dataset is **Random Forest** with an **RMSE** of **3523.17** and **MAPE of 9.67%.**

We had tried several models and tested each one of them on the dataset and found out how each model fared against the dataset.

We had first tried the **Linear Regression** Model and found out that it fared really well(**RMSE 3746.93**) on the dataset with One Hot Encoding as Data Pre-Processing rather than Label Encoding.

Then we tried **Polynomial Regression**(degree 2) which had an **RMSE** of **5760** with Label Encoding. With one hot encoding,a problem of memory error was coming due to the large number of columns formed by One Hot Encoding and also the huge computations of Polynomial Regression.

Next we tried **Support Vector Machines** which had an **RMSE** of **6811.54** with Label Encoding.Given the huge RMSE with this,we did not proceed with trying it with One Hot Encoding.

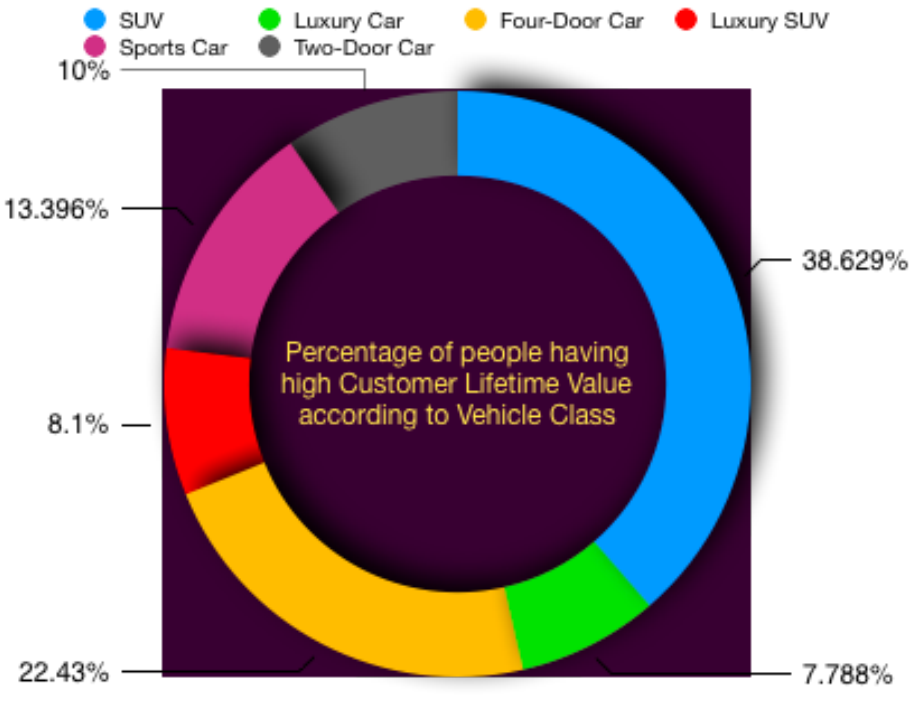
Then,finally we used **Random Forest Regression** Model on the dataset along with One Hot Encoding as Data Pre-Processing,which gave the lowest **RMSE 3523**. To aid the process,we also tuned the Hyper-Parameters which finally helped us to get this RMSE.

RECOMMENDATION & INSIGHTS

We have analyzed the data of people who have high Customer Lifetime Value and got the following insights.

We are first setting a threshold value of Customer Lifetime Value and then analysing the relation between the number of people having Customer Lifetime Value greater than that threshold value and different parameters :

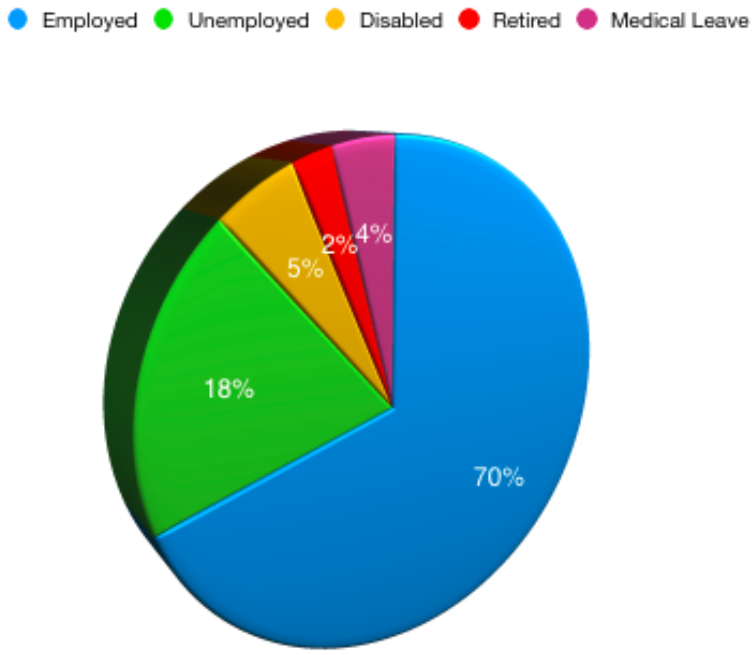
- **Vehicle Class :** We have plotted the pie chart of number of people who have high Customer Lifetime Value and the class of vehicle they own.



We can see from the graph that 38.6% people who have Customer Lifetime Value greater than 25000 have SUV as their vehicle.

So we suggest the company to focus on the customers who own SUV more as there is a higher chance of a consumer who owns a SUV to be more profitable for the company.

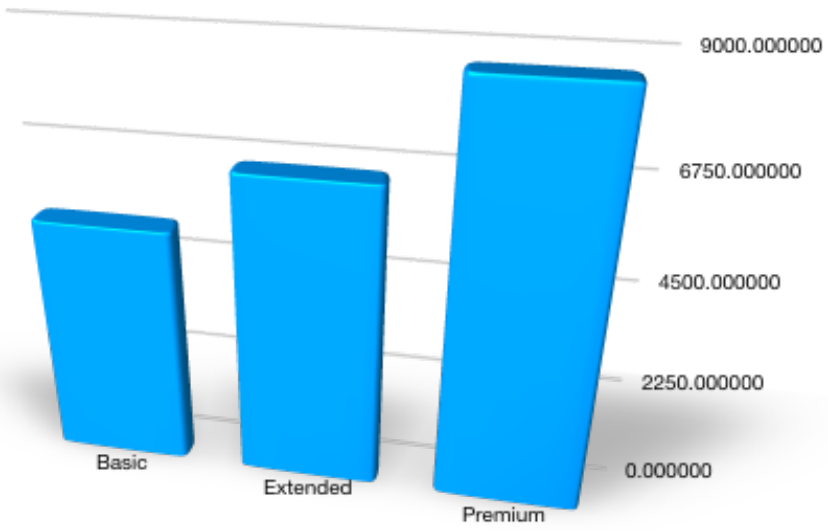
- **Employment Status :** We have plotted the pie chart of number of people who have high Customer Lifetime Value and their employment status.



From the graph, we can clearly see that people who are employed constitute 70% of the people who have high Customer Lifetime Value (CLV> 25000) .

Since employed people tend to be more profitable for the company, the company should offer employed people some exclusive discounts and offers. Also, the company should employ marketing techniques which target employed people.

- **Coverage :** We have plotted the chart of median of Customer Lifetime Value of people who have different coverage(Basic, Extended, Premium).



We see that the median of the CLV is higher for the people who have Premium Coverage.

The company should try to persuade people to take premium coverage by offering them some additional products, discount coupons, etc.

ANNEXURE

APPENDIX

SOFTWARE & LIBRARIES STACK

- **Python 3.6** - Language of choice
- **Pandas** - for Handling files
- **Numpy** - for complex numerical analysis
- **Seaborn** - plotting advance visualizations
- **Matplotlib** - plotting visualizations
- **Sklearn** - for making machine learning models

REFERENCES

https://pandas.pydata.org/pandas-docs/stable/
Pandas offer data structures and operations for manipulating numerical tables and time series.

https://docs.scipy.org/doc/
Numpy is a library used for computing scientific/mathematical data. Other usages are in:
1) Numerical Analysis 2) Linear algebra 3)Matrix computations

https://matplotlib.org/3.1.1/contents.html
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.
https://www.geeksforgeeks.org/ml-feature-scaling-part-1/
Feature scaling is the method to limit the range of variables so that they can be compared on common grounds. It is performed on continuous variables.

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
Train_test_split is helpful function for partitioning data which splits out your data into a training set and a test set.

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
It is used to show the linear relationship between a dependent variable and one or more independent variables.

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
Random Forest is an extension of bagging that in addition to building trees based on multiple samples of your training data, it also constrains the features that can be used to build the trees, forcing trees to be different. This, in turn, can give a lift in performance.

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html
Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning.

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

One-hot Encoding is a representation of categorical variables as binary vectors.This first requires that the categorical values be mapped to integer values.Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

Mean squared error is a risk function, corresponding to the expected value of the squared error loss. It is always non – negative and values close to zero are better. The MSE is the second moment of the error (about the origin) and thus incorporates both the variance of the estimator and its bias.