**Mahindra First Choice Capstone Project Approach**

**Team Name: Team Cyberbots**

**Team Members:**

Komal Bharadva

RuchitaSankhe

Ankita DuttaGupta

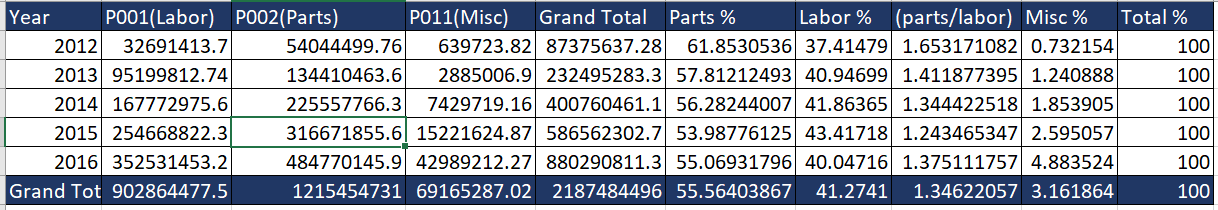
HeeraKesavan

**Exploratory Data Analysis for Capstone Project**

**Geolocation Based Analysis.**

**Parts to Labour Ratio:**

* Item Category P001 & P010 were considered for calculating labour charges
* Item Category P002 was considered for calculating part charges
* Item Category P011 was considered for calculating miscellaneous charges
* Grouping the merged data on Year & Item Category we got aggregated sum of the net value for each category
* Parts % = Revenue generated by parts charges(year) /Grand Total (year)
* Labour % = Revenue generated by labour charges (year) / Grand Total (year)
* Miscellaneous % = Revenue generated by Miscellaneous charges (year) / Grand Total (year)
* Parts to Labour Ratio = Parts Revenue(year) / Labour Revenue (year)

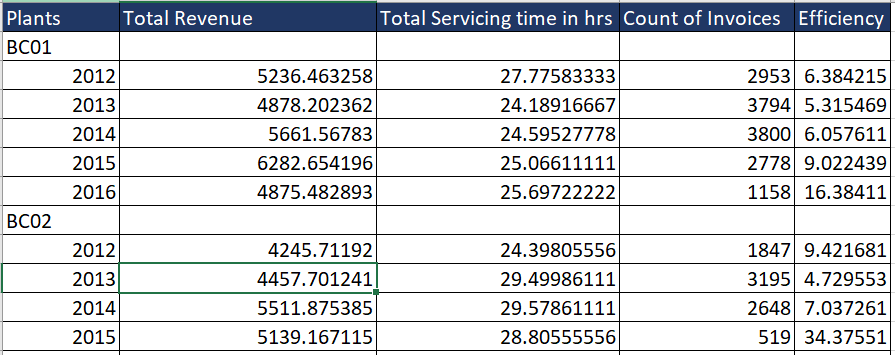


**Efficiency:**

* Efficiency was calculated Plant wise for each year & each month
* Service Time in Hrs was calculated by subtracting Invoice date time & job card date time columns
* Grouped the merged data on Plant, year& month and calculated Avg total revenue,

Average servicing time in hrs, Count of invoices.

* Efficiency = (Total Revenue in a year / Total servicing time in hrs \* Count of invoices) \*100



**Data Cleaning and Merging**

After initial analysis of missing values, we had decided to drop below features having missing values more than 40%

* **Invoice Data Cleaning**
  + Dropped Columns: - Amt Rcvd From Custom, Amt Rcvd From Ins Co,

CGST (14%), CGST (2.5%), CGST(6%),IGST(12%),CGST(9%),

IGST (18%), IGST (28%),IGST(5%),SGST/UGST(14%),

SGST/UGST (2.5%),SGST/UGST (6%),SGST/UGST(9%),

TDS amount,Service Advisor Name,Outstanding Amt,

Total CGST, TotalGST, Total IGST,Total SGST/UGST,

Policy no., Cash /Cashless Type,Expiry Date,Gate Pass Date,

Insurance Company, Claim No.,ODN No,Technician Name',

RegnNo',Total Value

* **Customer Data Cleaning**
  + Dropped columns: - Title, Marital Status, Occupation, Date of birth, Death date
* **Jtd Data Cleaning**
  + Dropped columns: -Labor Value Number
* **Data Merging**
  + Joined Customer & Invoice Data on Customer No. (inv\_cust)
  + Grouped the Jtd data on dbm order to reduce duplicate order entries (jtd\_group)
  + Left Joined the inv\_custwithgrouped Jtd data (jtd\_group) based on job card no &dbm order.
  + Final shape of merged dataframe (inv\_cust\_jtd)
  + Shape of joined dataframe (936275, 33)

**Feature Engineering**

* **Invoice Date Time**
  + Using Invoice date & invoice time **Invoice\_Date\_Time** was created
* **Job Card Date Time**
  + Using Jobcard date &Jobcard time **Jobcard\_Date\_Time** was created
* **Service Time**
  + By Subtracting invoice date time with Jobcard date time

**Feature Cleaning**

Following columns were cleaned using pgeocode library of python using pin codes given in Invoice data.

* **City**
  + Using pin code correct city was extracted
* **District**
  + Using pincode correct district was extracted
* **Area/Locality**
  + Using pincode correct location area was extracted

**Customer Segmentation**

After all cleaning & data pre-processing tasks, A new dataframe was created with the help of cleaned merged dataframe (inv\_cust\_jtd).

Grouping the merged data on Customer No, Cust Type, Make, Model, City, Order Type calculated average servicing time & average revenue & Count of invoices.

**Approach**

* Separating customers with zero spends
  + Separated customers having total revenue <= 0
  + 26327 rows were filtered as zero spenders
* Separating customers with nonzero spends
  + Separated customers having total revenue > 0
  + 284656 rows were filtered as non-zero spenders
* Rule Based Clustering on non-zero spenders using Average revenue
  + 4 clusters were identified based on quantile-based analysis on Average revenue.
  + Low Revenue, Medium Revenue, Average Revenue, High Revenue
* Rule Based Clustering on non-zero spenders using Average servicing time
  + 4 clusters were identified based on quantile-based analysis on Average servicing time in Hrs.
  + Super-Fast, Fast, Super Slow, Slow
* K-means Clustering on non-zero spenders
  + Label Encoded below columns
    - Make, Model, City, Cust Type, Order Type
  + Scaled all the features
    - Using standard scaler scaled all features
  + Used Elbow method to determine optimal no of clusters
    - We tried fitting range of 2 to 12 clusters on the scaled data.
    - Using elbow method, we found 6 clusters as optimal clusters
  + Run K-means clustering algorithm
    - 6 segments were generated by applying K-means algorithm with 6 clusters on scaled data.

**Customer Lifetime Value Prediction**

**Customer Lifetime Value**

Customer lifetime value is a metric that indicates the total revenue a business can reasonably expect from a single customer account. It considers a customer's revenue value, and compares that number to the company's predicted customer lifespan. Businesses use this metric to identify significant customer segments that are the most valuable to the company.

**Customer Lifetime Value Calculation**

CLTV = Customer frequency \* Average spend

**ApproachFor Classification**

* Created a Separate Data Frame for cars over a period of year2015
* On above DataFrame performed the group by on “Make”, “Model” “Location/Area” and “Customer No.” and perform the count on “Invoice No” (For the frequency of the particular Customer No.) , mean on “Service Time” (For average Service Time), mean on “Total Value” (For average purchase value)
* Multiply Count of Invoice No of individual customer to Average value of the respective customer to get the Customer Lifetime Value.
* Binned the Customer Lifetime value as “Low”, ‘Medium” and “High” to convert it into categorical format.
* Label Encode the Categorical columns.
* Drop columns "Customer No.","LTV (for Running Repairs)","Invoice No" as these columns are not required for Machine Learning Model building.
* Build a Machine Learning model to predict the classification of the customer in to categories of Customer Lifetime as “Low”, “Medium” and “High”. Use different Machine Learning and different features to try and improve the accuracy of the model.

**Approach For Regression**

* Created a Separate Data Frame for cars over a period of year 2015
* On above DataFrame performed the group by on “Make”, “Model” ,’Cust type’, ’Invoice no’, ’district’and “Customer No.” and perform the count on “Invoice No” (For the frequency of the particular customer no.),mean on “Total Amt Wtd Tax.” (For average purchase value)
* Multiply Count of Invoice No of individual customer to Average value of the respective customer to get the Customer Lifetime Value
* Creating Dataframe for Modelling with features ['Customer No.','Cust Type','district','Make','Invoice No','Total Amt Wtd Tax.','customer\_value']
* Renaming features as model\_df.columns=['Cust\_no.','Cust','State','Make','Count\_invoice','Avg\_revenue','customer\_value']
* Creating a dummy encoded data frame by applying pd.get\_dummies() on features[‘Cust’,Make,’State’]
* Finding the optimal value for Max\_depth
* Applying Decision Tree Regressor Algorithm on the features and evaluating the model on different evaluation metrics.

**Time series Forecasting using ARIMA model:**

* ‘Revenue’= Non Stationary data with respect to time series of 5 years
* Rolling Mean and Rolling standard deviation plot with a window of 12 months.
* Transformation of Data using 1st and 2nd order time differencing.

**Automated Dickey Fuller test**:

Null hypothesis: ‘data is non stationary’ .

Performed test on log shifted data

P value<0.05 : reject null hypothesis, Alternative hypothesis= data is stationary.

**Autocorrelation and Partial Autocorrelation function graph (ACF & PCF)**

Observations: presence of 1 autoregressive term, 1 moving average term.

* Time series model used : ARIMA of order (1,1,1)

p= 1, q= 1, d=1 from graphs

Result: RSS value= 4.76

**Prophet Model:**

* Prophet model is a facebook open source tool for business forecasting. It is a procedure for forecasting Time Series Data based on an additive model where non linear trends are fitted with yearly, weekly, daily seasonality.
* Prophet model only takes data as a dataframe with datestamp (ds) .
* Used " Invoice date" as x value and "Total amount with Tax" as y value.
* Created a forecast dataframe and specify the number of days using Periods parameter.  
   periods= 365 days
* Improved model by tuning using Changepoint and Seasonality Parameters.