A picture containing screenshot, laser, colorfulness, light

Description automatically generated

**Predictive Modeling Business Report**

V1.0

**Prepared by: Lavanya Sreeram**

**Prepared for:** GL Batch Nov 22

May 14, 2023

### Document Information

|  |  |
| --- | --- |
| **Author:** | Lavanya Sreeram |
| **Document Title** | Lavanya\_Sreeram\_14\_05\_2023 |
| **Issue Date:** | 14-05-2023 |
| **Version:** | V 1.0 |
| **Subject:** | Predictive Modeling Business Report by Lavanya Sreeram |

### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Revised By** | **Reason** |
| 1.0 | 14-05-2023 | Lavanya Sreeram | First Issue |
|  |  |  |  |
|  |  |  |  |

### Contents

[Document Information 1](#_Toc134991297)

[Revision History 1](#_Toc134991298)

[Contents 2](#_Toc134991299)

[1 Problem 1: Linear regression 9](#_Toc134991300)

[Data Description 9](#_Toc134991301)

[1.1 Exploratory Data Analysis 9](#_Toc134991302)

[1.1.1 Data Exploration 9](#_Toc134991303)

[1.1.2 Univariate Analysis: 11](#_Toc134991304)

[1.1.3 Bivariate Analysis 19](#_Toc134991305)

[1.2 Data Preparation 20](#_Toc134991306)

[1.2.1 Null Values 21](#_Toc134991307)

[1.2.2 Encode Data 22](#_Toc134991308)

[1.2.3 Outliers 23](#_Toc134991309)

[1.2.4 Co-relation 24](#_Toc134991310)

[1.2.5 Scaling 29](#_Toc134991311)

[1.3 Create Models 30](#_Toc134991312)

[1.3.1 Split Data 30](#_Toc134991313)

[1.3.1.1 VIF (Variance Inflation Factor) 31](#_Toc134991314)

[1.3.2 Model 1 – OLS Initial 32](#_Toc134991315)

[1.3.3 Model 2 – OLS Dropped p>.05 33](#_Toc134991316)

[1.3.4 Model 3 – OLS Scaled Data 34](#_Toc134991317)

[1.3.5 Model 4 – OLS Scaled and refined 35](#_Toc134991318)

[1.3.6 Model 5 – OLS Data – dropping variables 36](#_Toc134991319)

[1.3.7 Model 6 – Simple Linear Regression 38](#_Toc134991320)

[1.3.7.1 Actual v predicted scatter plot 39](#_Toc134991321)

[1.3.7.2 Residual plot 39](#_Toc134991322)

[1.3.8 Model 7 – Linear Regression on Scaled Data 40](#_Toc134991323)

[1.3.9 Model 8 – Linear Regresion on Scaled data- dropping features over VIF 5 41](#_Toc134991324)

[1.3.10 Scikit learn Linear Models Discussion & Equation 41](#_Toc134991325)

[1.3.11 Compare Scikit learn and OLS model – Model 5, Model 6 42](#_Toc134991326)

[1.4 Inference 42](#_Toc134991327)

[2 Problem 2: Logistic Regression, LDA, CART 43](#_Toc134991328)

[Data Description 43](#_Toc134991329)

[2.1 Exploratory Data Analysis 43](#_Toc134991330)

[2.1.1 Sample Data 44](#_Toc134991331)

[2.1.2 Data Statistical Description 44](#_Toc134991332)

[2.1.3 Missing Values 45](#_Toc134991333)

[2.1.4 Univariate Analysis 47](#_Toc134991334)

[2.1.5 Bivariate Analysis 60](#_Toc134991335)

[2.1.6 Encode data 63](#_Toc134991336)

[2.1.7 Outliers 66](#_Toc134991337)

[2.2 Prepare Data & Create Models 67](#_Toc134991338)

[2.2.1 Encode data 67](#_Toc134991339)

[2.2.2 Co-relation 67](#_Toc134991340)

[2.2.3 Split Data 69](#_Toc134991341)

[2.2.4 Model 1 – Decision Tree 70](#_Toc134991342)

[2.2.5 Model 2 – Apply Logistic, LDA, Cart 71](#_Toc134991343)

[2.2.6 Model 3 – Apply Logistic, LDA, CART on binned data 72](#_Toc134991344)

[2.3 Performance Metrics 78](#_Toc134991345)

[2.3.1 ROC Curve, Classification Report, Confusion Matrix 78](#_Toc134991346)

[2.3.2 Accuracy & ROC 84](#_Toc134991347)

[2.4 Inference 84](#_Toc134991348)

[2.4.1 Model Discussion 84](#_Toc134991349)

[2.4.2 Business Recommendations: 84](#_Toc134991350)

[3 Reflection Report: 86](#_Toc134991351)

Figures:

|  |  |  |
| --- | --- | --- |
| Figure No | Description | Pg No |
| 1.1.1 | Firm Data top 5 observations – as given | 10 |
| 1.1.2 | Firm Data bottom 5 observations – as given | 11 |
| 1.1.3 | Firm Data description - initial | 11 |
| 1.1.4 | Firm data after dropping ‘Unnamed:0’ feature | 11 |
| 1.1.5 | Firm data statistical description -initial | 12 |
| 1.1.6 | Histogram and countplot of the firm data | 20 |
| 1.1.7 | Firm data - Pairplot | 21 |
| 1.2.1 | tobinq’ boxplot | 22 |
| 1.2.2 | Firm Data Info after null values are treated | 23 |
| 1.2.2.1 | Firm Data after encoding | 23 |
| 1.2.2.2 | Firm Data after converting datatype | 23 |
| 1.2.3.1 | Firm Data boxplot showing outliers | 24 |
| 1.2.3.2 | Firm Data Boxplot after treating outliers | 25 |
| 1.2.4.1 | Heatmap – of all variables - before outliers | 26 |
| 1.2.4.2 | Heatmap – of independent variables - before outliers | 27 |
| 1.2.4.3 | Heatmap – of all variables - after treating outliers | 28 |
| 1.2.4.4 | Heatmap – of independent variables - after treating outliers | 29 |
| 1.2.5.1 | Firm Data after scaling | 30 |
| 1.3.1.1.1 | Variance Inflation Factor on ols train dataset | 32 |
| 1.3.1.1.2 | Variance Inflation Factor on scaled ols train dataset | 33 |
| 1.3.2 | Model 1 – OLS Initial Model | 34 |
| 1.3.3 | Model 2 – OLS Model dropped p>.05 | 35 |
| 1.3.4 | Model 3 – OLS Initial Model on scaled data | 36 |
| 1.3.5 | Model 4 – OLS Model on scaled data dropped p>.05 | 37 |
| 1.3.6 | Model 5 - OLS Improved model | 38 |
| 1.3.7.1 | Scatter plot -Actual v Predicted Sales for OLS Stats model | 40 |
| 1.3.7.2 | Residual Plot OLS Stats Model | 41 |
| 2.1.1.1 | Sample Crash Data top 5 | 45 |
| 2.1.1.2 | Sample Crash Data bottom 5 | 45 |
| 2.1.1.3 | Sample Crash Data top 5 -(Dropped unnamed, caseid) | 45 |
| 2.1.2.1 | Crash Data Description 1 | 46 |
| 2.1.2.2 | Crash Data Description 2 | 46 |
| 2.1.3.1 | Crash Data info-Missing values treated | 47 |
| 2.1.3.2 | Boxplot showing ‘injSeverity’ skewedness | 47 |
| 2.1.4 | histograms & Countplots of Crash data | 61 |
| 2.1.5.1 | Year Of Accident v Survival | 62 |
| 2.1.5.2 | Year of Vehicle on barplot with Survived as hue | 63 |
| 2.1.5.3 | Pairplot of Crash data | 64 |
| 2.1.6.1 | Crash data info after Encoding. | 65 |
| 2.1.6.2 | Crash data sample observation after Encoding | 65 |
| 2.1.6.3 | Crash data statistical description after Encoding | 66 |
| 2.1.7.1 | Boxplot showing Crash Data outliers | 67 |
| 2.1.7.2 | Boxplot showing Crash Data after treating outliers | 68 |
| 2.2.1.1 | Crash data info after Encoding. | 68 |
| 2.2.2.1 | HeatMap showing Crash data correlation | 69 |
| 2.2.3.1 | Crash data- Train Data sample | 70 |
| 2.2.3.2 | Crash data- Test Data sample | 70 |
| 2.2.3.3 | Crash data- Train Data sample after binning | 71 |
| 2.2.3.4 | Crash data- Test Data sample after binning | 71 |
| 2.2.4.1 | Model -1 Decision Tree basic | 71 |
| 2.2.4.2 | Importance of Features | 72 |
| 2.3.2 | Crash Data Regression Performance Metrics – Accuracy & ROC | 85 |

Tables:

Table1 VIF Range Page-33

# Problem 1: Linear regression

Predict the sales of these firms on the bases of the details given in the dataset so as to help your company in investing consciously. Also, provide them with 5 attributes that are most important.

## Data Description

[Firm\_level\_data.csv](https://olympus.mygreatlearning.com/courses/87096/files/7893754/download?wrap=1)[View in a new window](https://olympus.mygreatlearning.com/courses/87096/files/7893754/download?wrap=1) data set is provided. Data Dictionary for Firm\_level\_data is as below:

1. sales: Sales (in millions of dollars).
2. capital: Net stock of property, plant, and equipment.
3. patents: Granted patents.
4. randd: R&D stock (in millions of dollars).
5. employment: Employment (in 1000s).
6. sp500: Membership of firms in the S&P 500 index. S&P is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the United States
7. tobinq: Tobin's q (also known as q ratio and Kaldor's v) is the ratio between a physical asset's market value and its replacement value.
8. value: Stock market value.
9. institutions: Proportion of stock owned by institutions.

## Exploratory Data Analysis

Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, data types, shape, EDA). Perform Univariate and Bivariate Analysis. (8 marks)

Answer:

### Data Exploration

The Dataset has 759 rows, 10 columns. ‘sales’ is the Target Variable. The below snap shows the dataset first 5 observations.

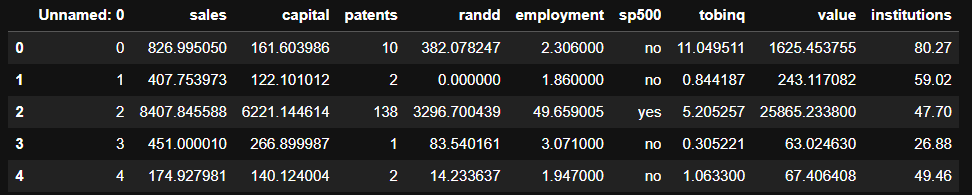


Fig1.1.1 Firm Data top 5 observations – as given

The below snap shows the bottom 5 observations.

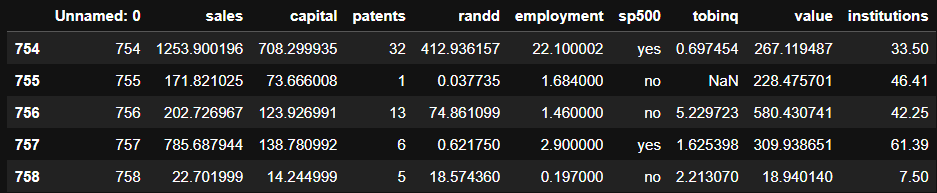


Fig1.1.2 Firm Data bottom 5 observations – as given

On first look, it can be inferred that the feature ‘Unnamed: 0’ must be dropped. Additionally, it can be noted that the data is not scaled.

The below snap shows the datatypes of all features of the dataset.

A screenshot of a computer program

Description automatically generated with low confidence

(Fig1.1.3): Firm Data description - initial

Except ‘sp500’, all the features in the data are numeric in nature ('int64' or 'float64' type). ‘sp500’ is object type, and as per data dictionary the feature is binary. ‘sp500’ shall be encoded.

Drop ‘Unnamed:0’ feature. Now, the dataset top 5 observations are as shown below:

A picture containing screenshot, text

Description automatically generated  
(Fig 1.1.4): Firm data after dropping ‘Unnamed:0’ feature

The statistical data descriptions is as shown in below snap.

A picture containing text, screenshot

Description automatically generated

(Fig1.1.5): Firm data statistical description -initial

### Univariate Analysis:

Univariate analysis on the dataset is as shown below using histograms:A graph of sales

Description automatically generated with low confidence

A picture containing text, screenshot, display, diagram

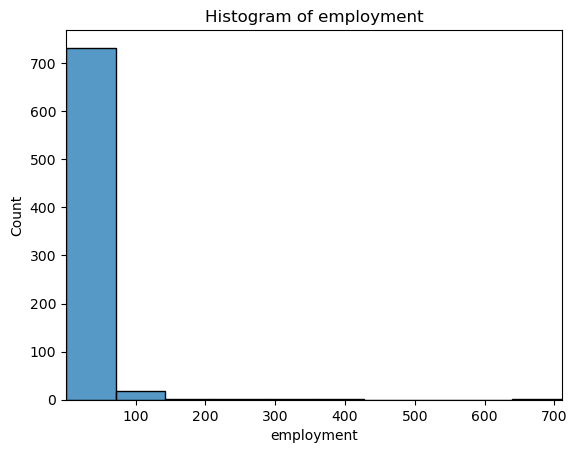
Description automatically generated

A picture containing text, screenshot, display, diagram

Description automatically generated

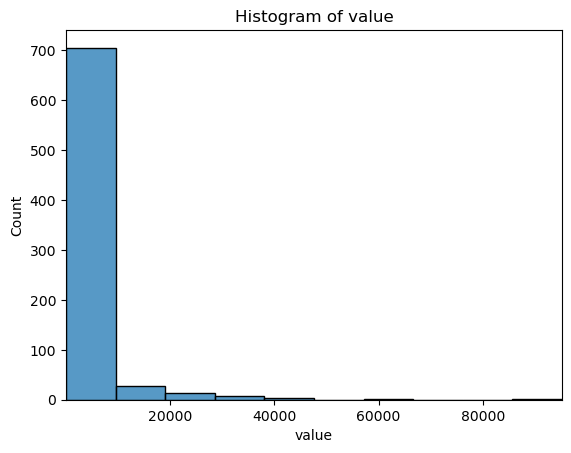
A picture containing text, screenshot, display, diagram

Description automatically generated

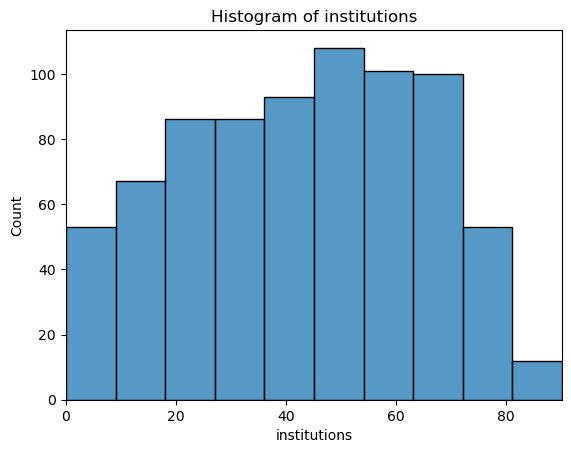


A picture containing text, screenshot, diagram, plot

Description automatically generated



All the above histograms show that the data is right skewed or positively skewed meaning that the mean is greater than the median. This is because the presence of a few large values on the right pulls the mean towards higher values. The tail of the distribution extends towards larger values, indicating the presence of outliers or extreme values on the right side. This can happen as few observations in ‘sales’ or ‘value’ are significantly higher in weight/value compared to the majority. This is generally the case with economic observations like stock prices.



From the above histogram, it can be noted that the ’institutions’ is not as skewed as rest of the features. In fact, the feature is more uniform and has left skewedness in data.

Count plot is used for the binary variable ‘sp500’ as shown below:

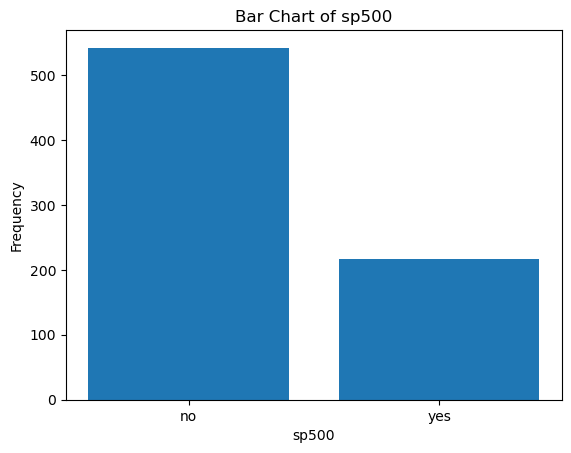
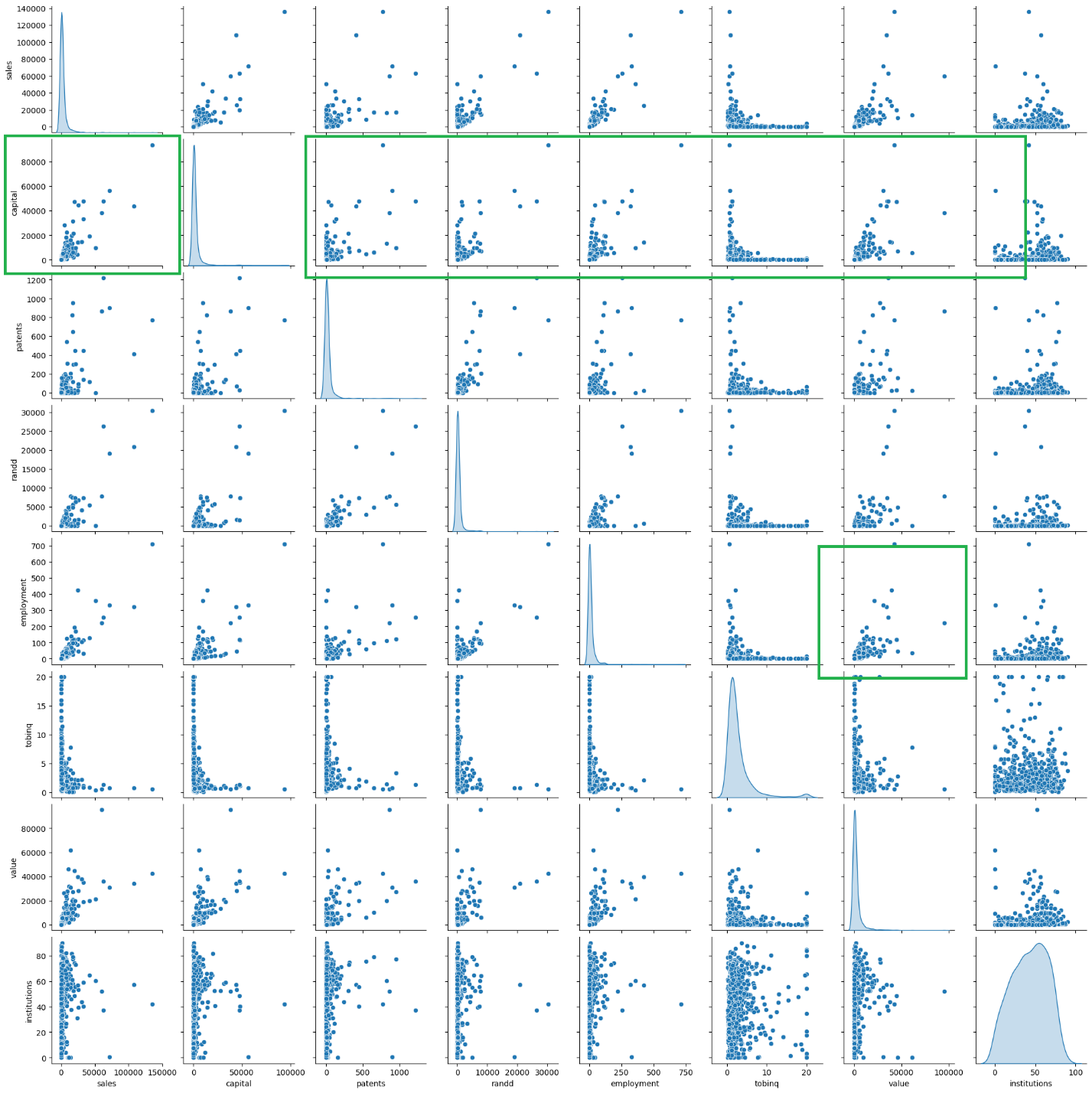


Fig1.1.6 Histogram and countplot of the firm data

The plot shows that the observations with S&P 500 index companies are less in number compared to opposite.

### Bivariate Analysis

Bivariate analysis is performed on the data and the pair plot is shown below:  
Fix- highlight co-relation features.

  
(Fig 1.1.7): Firm data – Pairplot

There is lot of positive high correlation & linear relationship for ‘capital’ variable with multiple variables. Similar insights are drawn from the heatmap.

## Data Preparation

Impute null values if present? Do you think scaling is necessary in this case? (8 marks):   
  
Answer:

### Null Values

There are no duplicates in the data. There are 21 null/missing values in ‘tobinq’ column. Boxplots are generated to visualize the skewness in data as shown below.

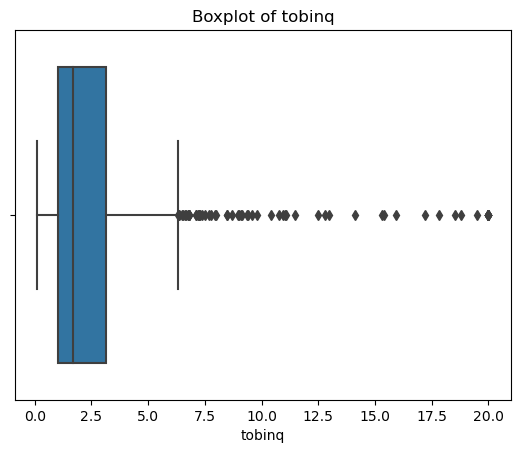


Fig1.2.1 ‘tobinq’ boxplot

The data is right skewed. The missing values in ‘tobinq’ are treated by median imputation. Median imputation is preferred when the distribution is skewed, as the median is less sensitive to outliers than the mean.

After treating the null values in the data, the data info can be seen as below with no null values.

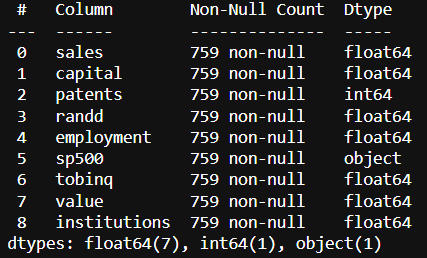


Fig1.2.2 Firm Data Info after null values are treated

### Encode Data

The ‘sp500’ feature is the only categorical feature and contains binary values. After one-hot encoding, the dataset looks like below.

A picture containing screenshot, text, software

Description automatically generated

Fig1.2.2.1 Firm Data after encoding

The dataset types are as below after the encoding:

A screenshot of a computer program

Description automatically generated with medium confidence

Fig1.2.2.2 Firm Data after converting datatype

### Outliers

The boxplot is plotted to visualize the outliers in all numerical features i.e. except ‘sp500’ which is essentially binary/categorical feature.

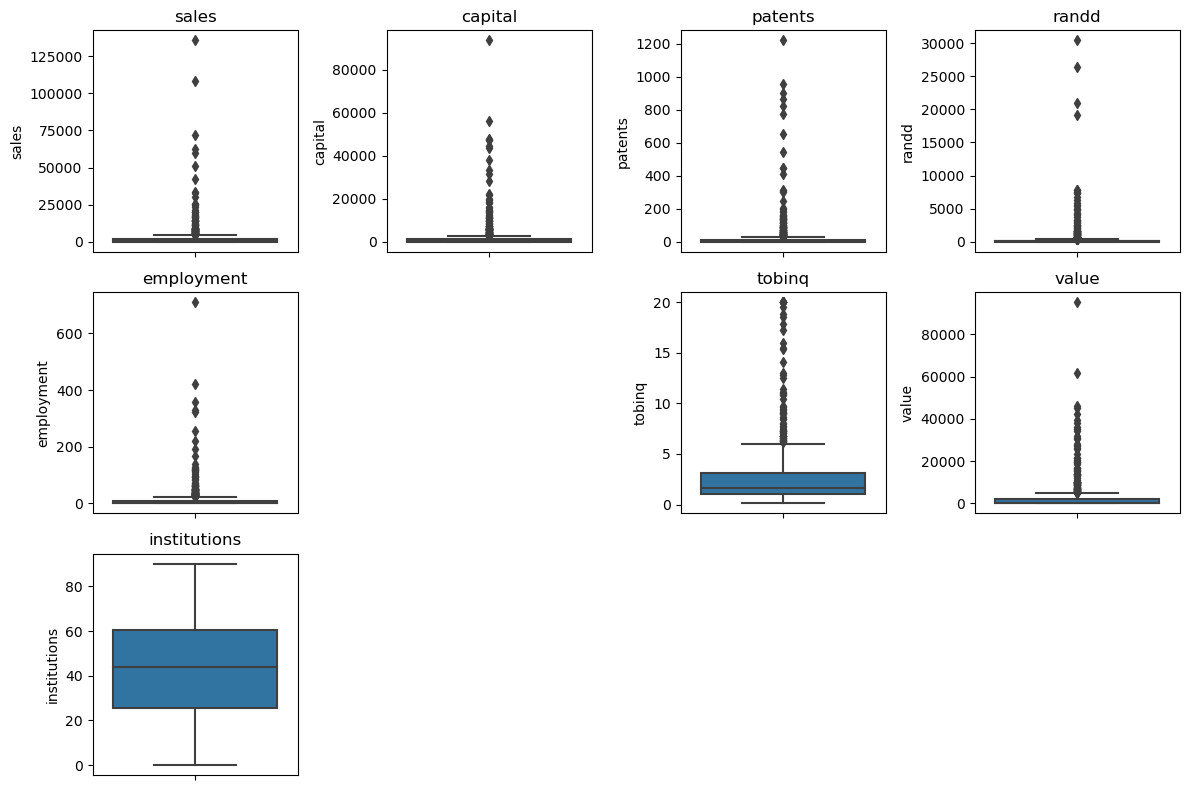


Fig1.2.3.1 Firm Data Boxplot showing outliers

There are multiple outliers in many columns as shown in above plot.

The black dots in the boxplots show that there are multiple outliers in multiple columns. Except 'institutions' remaining features have outliers. Majority of the variables are highly skewed as well and this can be seen in the boxplots as significantly larger tails with large magnitude (Y value).

Treat outliers:

All the outliers are treated by adjusting them to the lower and upper bound values calculated by the IQR value.

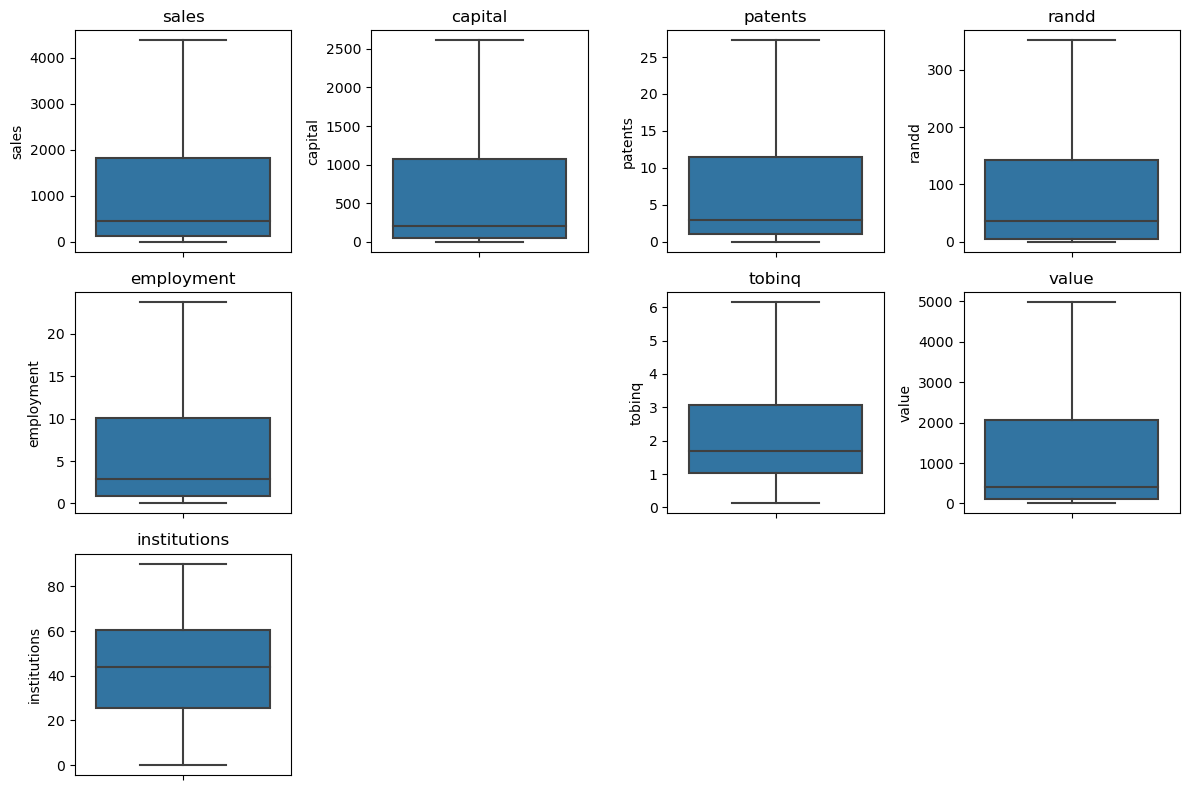


Fig1.2.3.2 Firm Data Boxplot after treating outliers

The above boxplot is after treating the outliers showing no outliers.

### Co-relation

4 co-relation heat maps are shown in this section. These are heatmaps of independent features , heatmaps of all features before and after treating outliers.

Before treating outliers:

The below plot shows the correlation between all variables before outliers are treated:

A picture containing text, screenshot, rectangle, square

Description automatically generatedFig1.2.4.1 Heatmap – of all variables  
The below plot shows the correlation between independent variables before outliers are treated.  
A picture containing text, screenshot, rectangle, square

Description automatically generated

Fig1.2.4.2 Heatmap – of independent variables - before outliers

Below chart shows there is lots of multi-collinearity and co-relation between variables after treating the outliers.

A picture containing screenshot, square, rectangle, pattern

Description automatically generated

Fig1.2.4.3 Heatmap – of all variables - after treating outliers

The below heatmap shows the co-relation matrix of independent variables:

A picture containing screenshot, square, rectangle, pattern

Description automatically generated

Fig1.2.4.4 Heatmap – of independent variables - after treating outliers

Based on the above correlation matrix plot:

There are several strong and moderate positive correlations, indicating that certain variables tend to move together in a positive direction. For example, ‘capital’ shows strong positive correlations with ‘patents’, ‘randd’, and ‘employment’.

'tobinq', 'institutions' are weakly co-related. 'patents', 'randd', 'sp500' are moderately co-realted with values around .6/.7. 'capital', 'employment', 'value' are highly co-related with correlation values over .91.

'value' shows moderate positive correlations with multiple variables, indicating potential associations with those variables. The variable 'institutions' generally has weak positive correlations with other variables.

The variable 'tobinq' has weak negative correlations with several other variables, suggesting a weak inverse relationship.

There is high co-relation between the independent variables. This needs to be addressed. It's important to note that correlation alone does not imply causation. Further analysis and domain knowledge are needed to determine the underlying relationships and causality between variables.

### Scaling

The below snap shows the data description after the data is prepared.  


Fig1.2.5.1 Firm Data after scaling

Scaling - Magnitude Differences:

The variables in the dataset have significantly different scales. For example, 'sales' and 'value', 'capital' have much larger magnitudes compared to 'patents', 'employment', and 'institutions'. Scaling can help to ensure that all features contribute equally to the model.

That being said, it is not necessary to perform scaling on the data. In the later sections, ols regression model and linear regression model are fit on the original and scaled data. The models show no major difference in terms of performance metrics.

## Create Models

Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (30:70). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using R-square, RMSE. (8 marks)

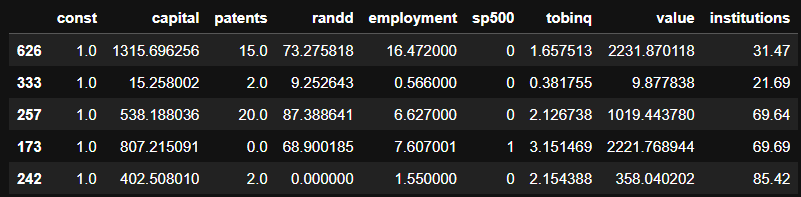
Answer:

### Split Data

The ‘sales’ feature is the target variable, y. Rest of the data is X. Using sklearn train\_test\_split feature, the data X and y is split into train and test sets in a 70:30 ratio.

Train Test Data for the Stats Model:

The Train dataset top 5 rows are shown below.



The test dataset top 5 rows are shown below.

A picture containing text, screenshot, font

Description automatically generated

The training and test dataset sample for stats model **after scaling** is shown below:

A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font, software

Description automatically generated

#### VIF (Variance Inflation Factor)

The below snap the sorted features by Importance of the sm ols original train dataset.



Fig1.3.1.1.1 Variance Inflation Factor on ols train dataset

Below snap shows the sorted features by VIF importance of the sms ols scaled train dataset.



Fig1.3.1.1.2 Variance Inflation Factor on scaled ols train dataset

In both cases, ‘employment’, ‘capital’, ‘value’ have high multicollinearity with VIF value > 5. However, dropping variables blindly based on VIF value may not always result in better models.

Using VIF, features with high co-linearity can be identified. For reference, the range of VIF:

|  |  |
| --- | --- |
| VIF | Starts at 1 |
| VIF = 1 | no correlation between this independent variable and the other variables |
| 1 < VIF < 5 | Indicates moderate collinearity |
| VIF > 5 | exceeding 5 or 10 indicates high multicollinearity between this independent variable and the other. |

Table1 VIF Range

### Model 1 – OLS Initial

OLS Stats Model is fit on the train data. The performance metrics on the initial model are shown as below.

Initial OLS Model Training Set Metrics:

RMSE: 394.33716619028996

R-squared: 0.9359702538559448

Initial OLS Model Test Set Metrics:

RMSE: 400.0021149372728

R-squared: 0.9240311293641787

The summary of the results is as shown below:

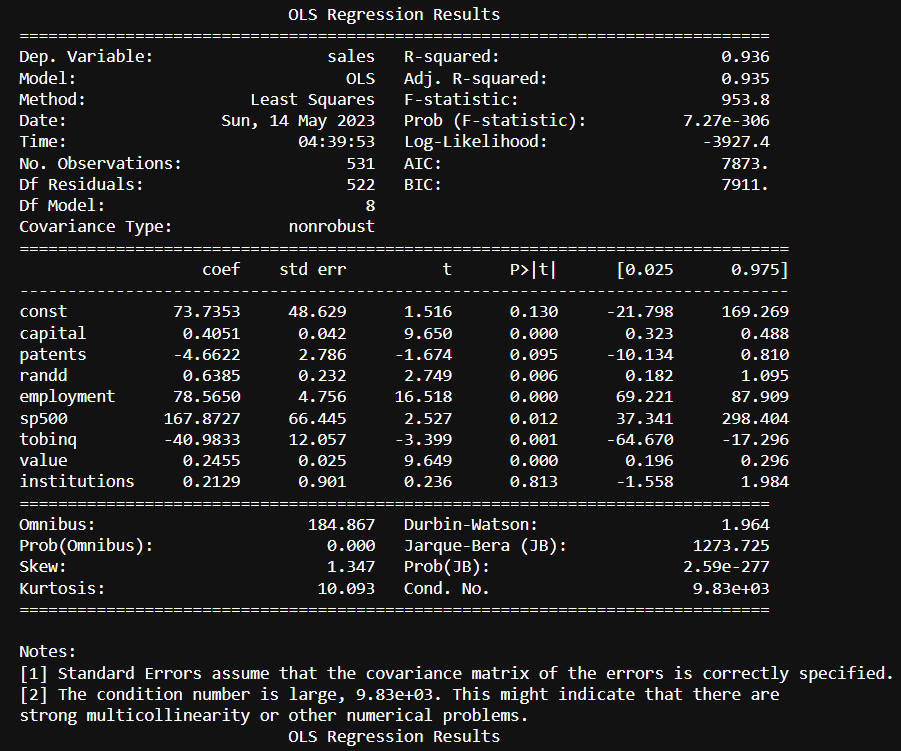
The OLS Regression results are as below:

Fig1.3.2 Model 1 – OLS Initial Model

### Model 2 – OLS Dropped p>.05

All features with p values above .05 are dropped and the new model regression performance metrics are:

Training Set Metrics:

RMSE: 395.3975671797366

R-squared: 0.9356254296687838

Test Set Metrics:

RMSE: 398.4877494359508

R-squared: 0.9246052606459508

And the summary of result is as shown below.

A screenshot of a computer

Description automatically generated

Fig1.3.3 Model 2 – OLS Model dropped p>.05

The model assumptions, such as normality and absence of autocorrelation, should be carefully evaluated.

### Model 3 – OLS Scaled Data

The performance metrics on the OLS Scaled data are as below:  
  
Scaled Model Training Set Metrics:  
RMSE: 394.33716619028996  
R-squared: 0.9359702538559448

Scaled Model Test Set Metrics:  
RMSE: 400.002114937273  
R-squared: 0.9240311293641786

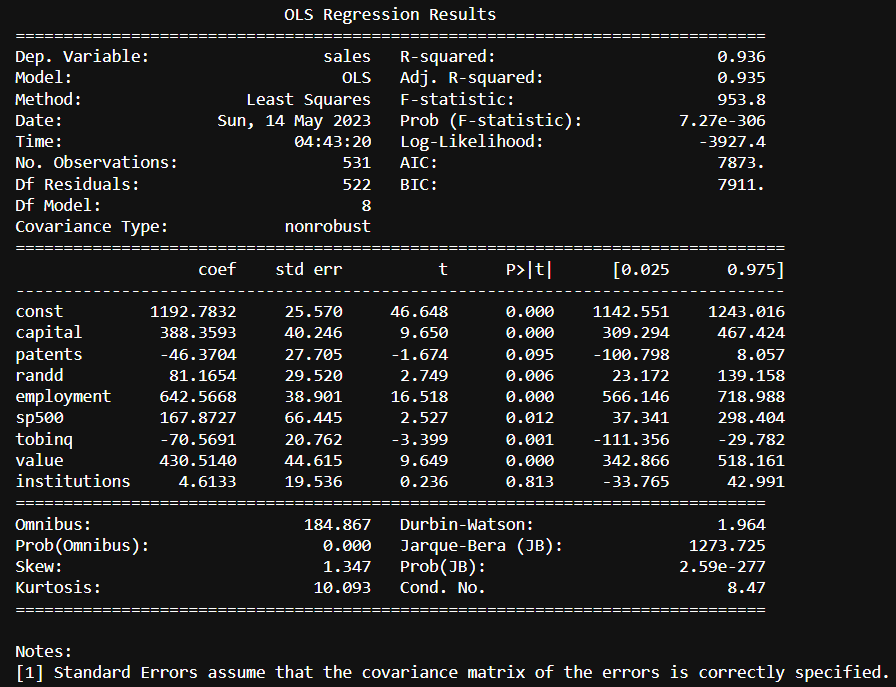
regression result on scaled data are as shown below:  


Fig1.3.4 Model 3 – OLS Initial Model on scaled data

### Model 4 – OLS Scaled and refined

The scaled and new model after dropping all features with p>0.05 has below metrics:  
Scaled Improved Model Training Set Metrics:

RMSE: 395.3975671797366

R-squared: 0.9356254296687838

Scaled Improved Model Test Set Metrics:

RMSE: 398.48774943595146

R-squared: 0.9246052606459505

The complete summary is as shown below:

A screenshot of a computer screen

Description automatically generated with medium confidence

Fig1.3.4 Model 4 – OLS Model on scaled data dropped p>.05

### Model 5 – OLS Data – dropping variables

Blindly dropping all features above p>0.05 is not always correct as the dropped features may have multi-coliinearity but still carry important information about the target variable. So instead, drop each variable based on p-value, lower rmse, higher r-squared, Higher F-Statistic. This is iterative process and in the final improved model 'sales','patents','institutions','randd' are dropped.

The OLS Stats improved regression model equation can be given as below:

y = (81.7013 \* const) + (0.4118 \* capital) + (80.6845 \* employment) + (176.4655 \* sp500) - (39.6549 \* tobinq) + (0.2466 \* value)

1. When all other predictors are zero, the const is 81.703 which is not bad.
2. for a one-unit increase in the "sp500" variable, holding all other variables constant, the sales price increases by 176.4655
3. For one-unit increase in ‘tobinq’ the sales decreases by -39.6549 as the ‘-‘ sign indicates an inverse relationship.

The performance metrics and the summary of this improved ols model are as below:

Training Set Metrics:

RMSE: 397.20671726067104

R-squared: 0.935034987496708

Test Set Metrics:

RMSE: 394.38620894692014

R-squared: 0.9261493138182268

Performance metrics on test data-

1. An R-squared value of 0.926 suggests that the model explains approximately 92.6% of the variance in the sales data.
2. The RMSE for the test set indicates the average deviation of the predicted sales values from the actual sales values in the test set. A lower RMSE is desirable, indicating better predictive performance. The RMSE of 394.38 suggests an average deviation of approximately 398.488 which is pretty good considering the target variable range.
3. The target variable has a mean of 2689.705158 and a standard deviation of 8722.060124. In this context, an RMSE of 394 is relatively small compared to the magnitude of the target variable. It indicates that, on average, the model's predictions have an error of approximately 394 units, which is relatively low compared to the overall scale of the target variable. Hence, this is an acceptable value of RMSE.

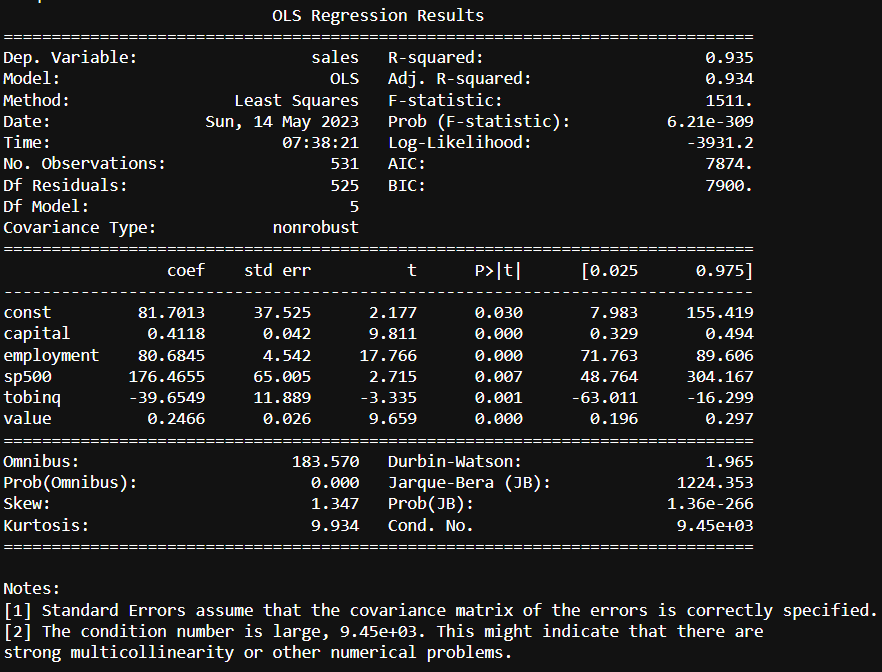


Fig1.3.6 Model 5 - OLS Improved model

### Model 6 – Simple Linear Regression

Below are the Lienar regression results:

Coefficients: [[ 0.40506319 -4.66215536 0.63853807 78.56497068 167.87266108

-40.98329451 0.24554026 0.21287606]]

Intercept: [73.73534908]

----

Training set performance:

RMSE: 394.337

R-squared: 0.936

MSE 155501.80063898838

Test set performance:

RMSE: 400.002

R-squared: 0.924

MSE 160001.69195429183

#### Actual v predicted scatter plot

A picture containing text, screenshot, diagram, line

Description automatically generated

Fig1.3.7.1 Scatter Plot- Actual v Predicted Sales for OLS Stats model.

The above scatter plot shows between actual and predicted values. Inferences are:

1. There are 3 areas, one where the sales are below ‘1000’, between ‘1000-2000’ & higher.
2. The model correctly predicts level 1, whereas the variance increases as the sales value increases.
3. The variance of the error is not constant across various levels of your dependent variable.

#### Residual plot

A picture containing text, screenshot, diagram, line

Description automatically generated

Fig1.3.7.2 Residual Plot OLS Stats Model

1. The residual points are symmetrically distributed, tending to cluster towards the middle of the plot.
2. They’re clustered around the lower single digits of the y-axis.
3. In general, there aren’t any clear patterns.
4. This is an ideal residual plot and is acceptable.

Note- Scatter plots and residual plots for all models have been plotted and all show similar result.

### Model 7 – Linear Regression on Scaled Data

Model performance results are:

Coefficients: [[388.35931937 -46.37038134 81.16536349 642.56680013 167.87266108

-70.56912892 430.5139761 4.61330005]]

Intercept: [1192.78318419]

----

Training set performance:

RMSE: 394.337

R-squared: 0.936

MSE 155501.8006389884

Test set performance:

RMSE: 400.002

R-squared: 0.924

MSE 160001.69195429125

### Model 8 – Linear Regresion on Scaled data- dropping features over VIF 5

Model performance results are:

Coefficients: [[ 21.6288429 429.0632531 2050.9331315 -246.32957 54.98948947]]

Intercept: [669.99885843]

----

Training set performance:  
RMSE: 854.696

R-squared: 0.699

MSE 730504.8254082266

Test set performance:

RMSE: 948.881

R-squared: 0.573

MSE 900375.8143274568

### Scikit learn Linear Models Discussion & Equation

Below is the discussion on Scikit learn’s linear models:

1. As discussed above, and based on the Model 6,7,8 results, scaling does not necessary give drastic results in this dataset. The Model 6, 7 have similar or same performance result upto or above 5 decimal points.
2. The best linear model is Model 6 – Simple Linear regression without scaling.
3. Linear regression can be given by below equation (Model 6 – Simple Linear Regression):

y = 0.40506319 \* capital - 4.66215536 \* patents + 0.63853807 \* randd + 78.56497068 \* employment + 167.87266108 \* sp500 - 40.98329451 \* tobinq + 0.24554026 \* value + 0.21287606 \* institutions + 73.73534908

1. Dropping features, blindly based on VIF value may not give better model. Infact, as seen in model 8, the model may show decrease in predicting performance.

### Compare Scikit learn and OLS model – Model 5, Model 6

1. Scikit learn gives Test set performance of RMSE: 400.002, R-squared: 0.924
2. Stats model gives test set performance of RMSE: 394.38620894692014, R-squared: 0.9261493138182268
3. Both models are strong models for regression. However, Stats model is slightly a better model compared to scikitlearn.

## Inference

Based on these predictions, what are the business insights and recommendations. (6 marks)

1. There might still be multicollinearity which needs to be addressed with business team.
2. Stats model is used for model prediction given equation y = (81.7013 \* const) + (0.4118 \* capital) + (80.6845 \* employment) + (176.4655 \* sp500) - (39.6549 \* tobinq) + (0.2466 \* value)
3. ‘capital’, ‘employment’, ‘tobinq’, ‘value’, ‘sp500’ are the best predictors of the target variable.
4. ‘institutions’, ‘patents’, ‘raandd’ do not show a significance value in predicting the target variable. However, this does not mean they are poor predictors.
5. Focus on Capital Investment - as this variable has a high positive impact on ‘sales’
6. Monitor S&P 500 Performance: The coefficient of the "sp500" variable is 176.4655. This indicates that the performance of the S&P 500 index has a positive impact on the response variable. It is recommended to closely monitor the trends and movements of the S&P 500 index as it can potentially influence the ‘sales’ outcome.
7. ‘tobinq’ has an inverse relationship and the company should evaluate this variable carefully as this has a negative effect on the ‘sales’.

# Problem 2: Logistic Regression, LDA, CART

You are hired by the Government to do an analysis of car crashes. You are provided details of car crashes, among which some people survived and some didn't. You have to help the government in predicting whether a person will survive or not on the basis of the information given in the data set so as to provide insights that will help the government to make stronger laws for car manufacturers to ensure safety measures. Also, find out the important factors on the basis of which you made your predictions.

## Data Description

Data Dictionary for Car\_Crash

Dataset for Problem 2: [Car\_Crash.csv](https://olympus.mygreatlearning.com/courses/87096/files/7893755/download?wrap=1)[View in a new window](https://olympus.mygreatlearning.com/courses/87096/files/7893755/download?wrap=1)

1. dvcat: factor with levels (estimated impact speeds) 1-9km/h, 10-24, 25-39, 40-54, 55+  
2. weight: Observation weights, albeit of uncertain accuracy, designed to account for varying sampling probabilities. (The inverse probability weighting estimator can be used to demonstrate causality when the researcher cannot conduct a controlled experiment but has observed data to model)  
3. Survived: factor with levels Survived or not\_survived  
4. airbag: a factor with levels none or airbag  
5. seatbelt: a factor with levels none or belted  
6. frontal: a numeric vector; 0 = non-frontal, 1=frontal impact  
7. sex: a factor with levels f: Female or m: Male  
8. ageOFocc: age of occupant in years  
9. yearacc: year of accident  
10. yearVeh: Year of model of vehicle; a numeric vector  
11. abcat: Did one or more (driver or passenger) airbag(s) deploy? This factor has levels deploy, nodeploy and unavail  
12. occRole: a factor with levels driver or pass: passenger  
13. deploy: a numeric vector: 0 if an airbag was unavailable or did not deploy; 1 if one or more bags deployed.  
14. injSeverity: a numeric vector; 0: none, 1: possible injury, 2: no incapacity, 3: incapacity, 4: killed; 5: unknown, 6: prior death  
15. caseid: character, created by pasting together the populations sampling unit, the case number, and the vehicle number. Within each year, use this to uniquely identify the vehicle.

## Exploratory Data Analysis

Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. (8 marks)

Answer:  
The data shows 11217 rows, 16 columns. ‘Survived’ is the target or feature of interest with ‘survived’ having higher value count and hence the positive class.

### Sample Data

The Data provided has these top rows as shown below.

A picture containing screenshot, text

Description automatically generated

Fig2.1.1.1 Sample Crash Data top 5

The bottom five rows from the dataset are shown below.

A screenshot of a computer

Description automatically generated with low confidence

Fig2.1.1.2 Sample Crash Data bottom 5

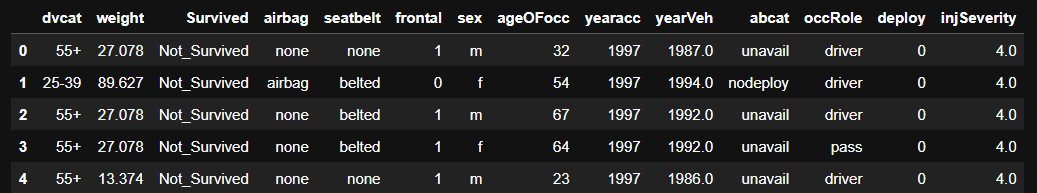
The data contains ‘Unnamed: 0’, ‘caseid’ features which are dropped. The dataset top 5 observations are as shown below:  


Fig2.1.1.3 Sample Crash Data top 5-(dropped unnamed, caseid)

The data contains many numerical and categorical columns. Further exploratory data analysis is as below.

### Data Statistical Description

The data description of categorical and numerical is shown below.

A picture containing text, screenshot, font

Description automatically generated

Fig2.1.2.1 Crash Data Description 1

A picture containing text, screenshot, font, number

Description automatically generated

Fig2.1.2.2 Crash Data Description 2

### Missing Values

Missing values:

Below information shows the missing values in the data.

A screen shot of a computer

Description automatically generated with low confidence

Fig2.1.3.1 Crash Data info-Missing values treated

There are 77 missing values in ‘injSeverity’ column. Below is boxplot of ‘injSeverity’:

A picture containing screenshot, diagram, text, rectangle

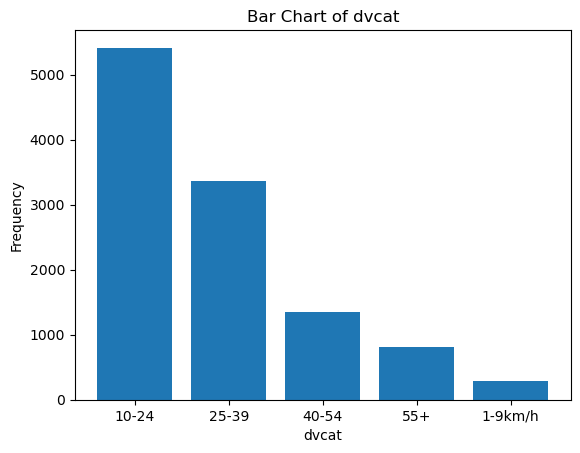
Description automatically generated

Fig2.1.3.2 Boxplot showing ‘injSeverity’ skewedness.

the 'injSeverity' does not have outliers. Data is very lightly right skewed. Hence, the missing values can be imputed based on median. Note- Mean can also be used as there is no high skewedness in data.

### Univariate Analysis

The univariate analysis is as shown below using histogram plots of each features:

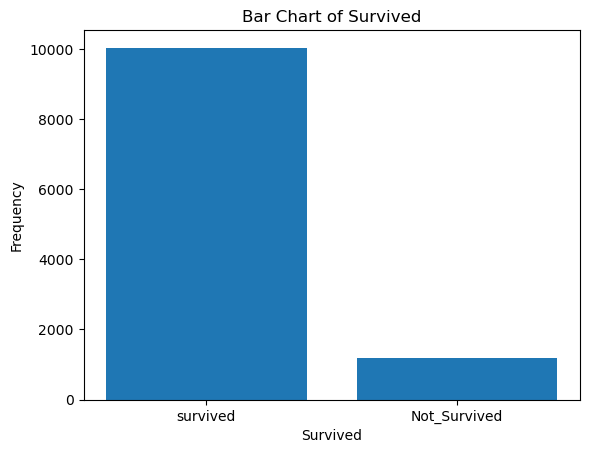


The dvcat is highly right skewed indicating outliers presence.

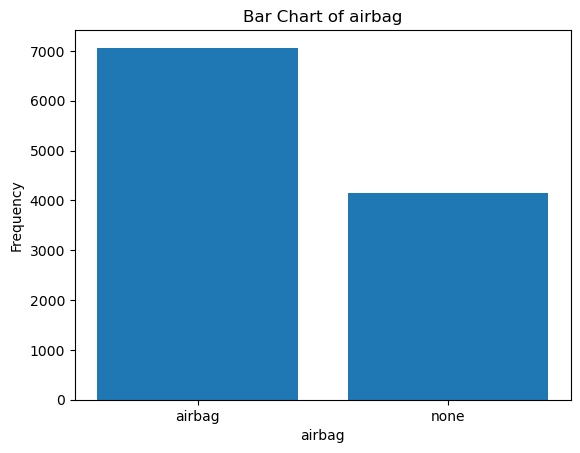
A picture containing text, screenshot, display, diagram

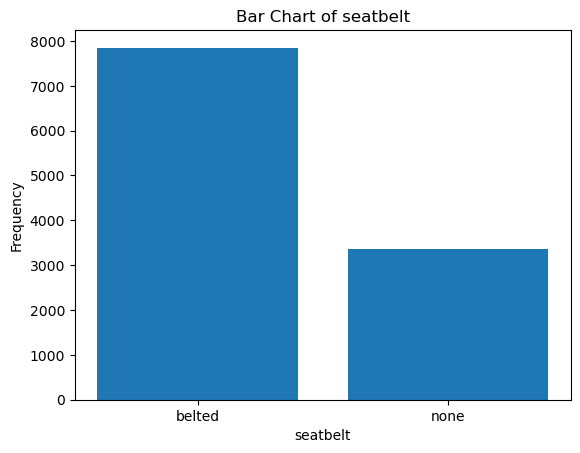
Description automatically generated

The ‘weight’ is highly right skewed indicating outliers presence.



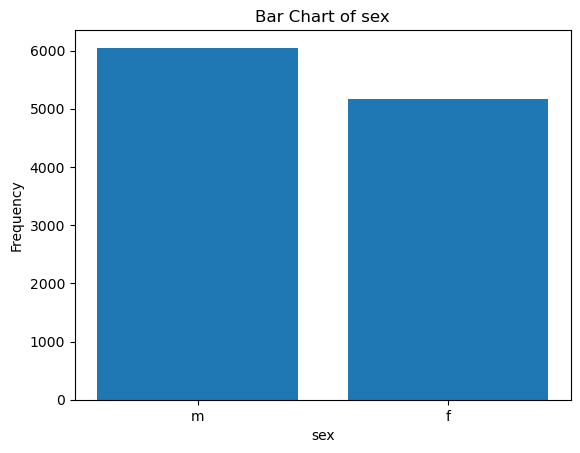
‘Survived’ i.e. the target variable has much higher counts for ‘Survived’ than the opposite and while splitting data stratify parameter shall be used.

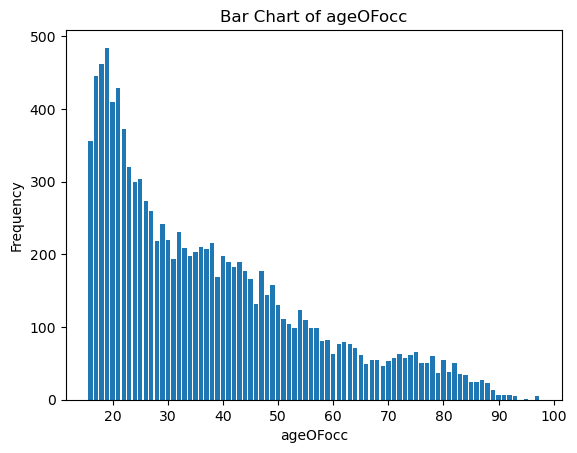




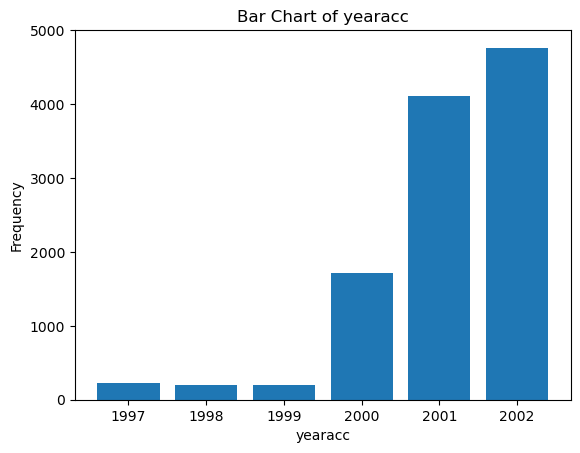
A picture containing text, screenshot, diagram, plot

Description automatically generated

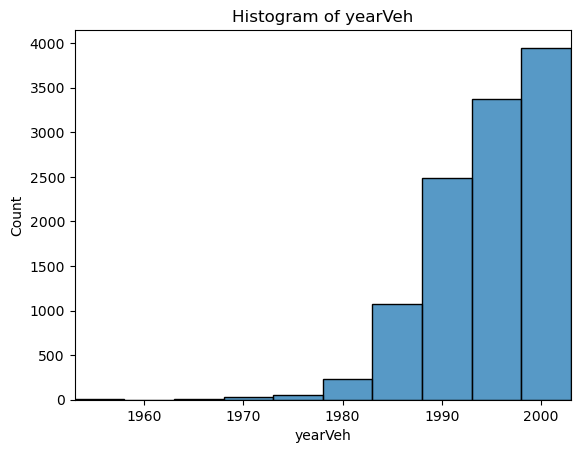




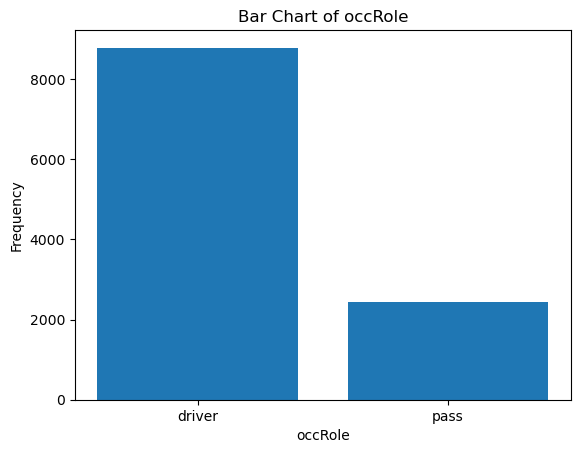
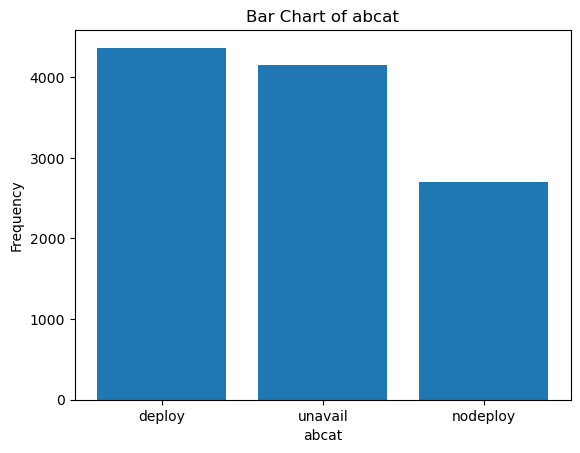
The ‘ageOfOcc’ has a peak in ’10-24’ age group. The data is also highly right skewed indicating the presence of ‘outliers’. Binning of age groups can help in this feature.

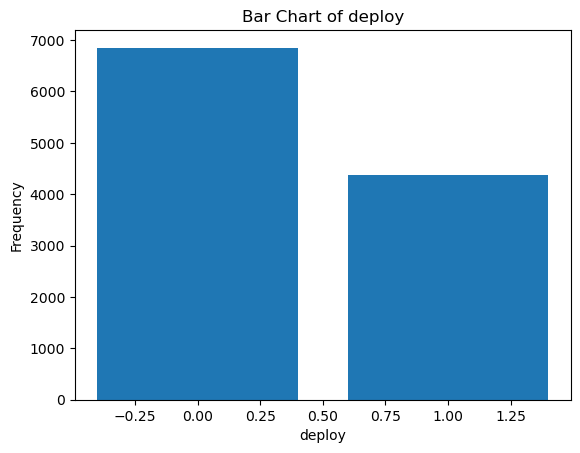


There seem to a lot more accidents in years after 2000. This can be due to many reasons that there are higher sales in cars/a boom in teenage car ownership/ records being maintained after these years. More analysis is needed to bring out more insights in this feature. Additionally, binning can help this feature as well.



‘yearVeh’ has highly left skewed data with most cars being from years after 1990. It can be due to increase of availability of these cars in the market from manufacturing point of view. More analysis is needed to bring out more insights in this feature. Additionally, binning can help this feature as well.





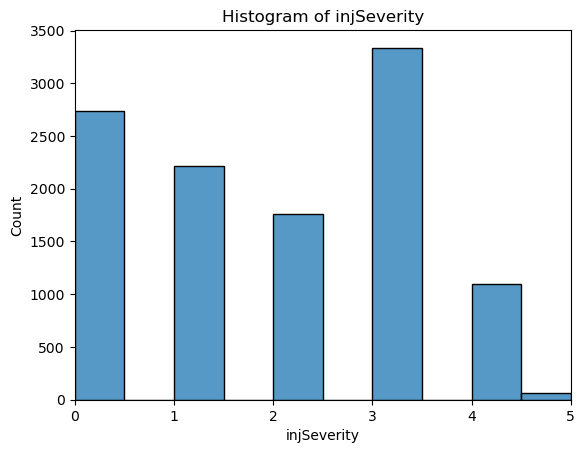
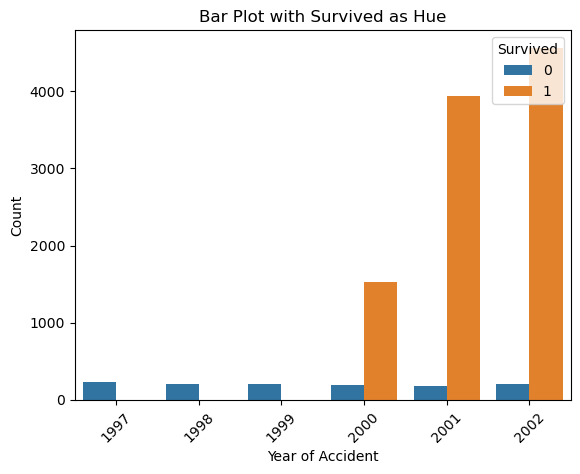


Fig2.1.4 histograms & Countplots of Crash data

The ‘injSeverity’ of 5 is very less and that can explain the high ‘NotSurvived’ count. More insights need to be drawn.

### Bivariate Analysis

Using calculation, we can find that The vehicles from 1990-2000 are involved in 61% and vehicles after 2000 account for 18% of the crashes.

  
Fig2.1.5.1 Year Of Accident v Survival

The above plot shows a clear distinction that accidents before 2000s resulted only in 'NotSurvived'. Based on this information, the year of accident is encoded into binary variable i.e. before 2000s as 0, after 2000s as 1

A picture containing text, screenshot, diagram, plot

Description automatically generated  
Fig2.1.5.2 Year of Vehicle on barplot with Survived as hue

Above plot shows that Compared to older version cars, passengers/drivers using latest version cars have drastically higher chance of survival. This can be attributed to the airbag facilities in latest cars

Bivariate analysis using pairplot is shown below:



Fig 2.1.5.3 Pairplot of Crash data

It can be seen from the pairplot as well that accidents before 2000s resulted only in 'NotSurvived'.The classes are not well separated as peaks are engulfed by other class.

### Encode data

The datatype of each feature is shown below:

A screen shot of a computer

Description automatically generated with medium confidence  
Fig 2.1.6.1 Crash data info after Encoding.

Inference: Out of 14 columns 7 are type object, rest are either of type float or integer. Most of the existing *numerical* columns are of type float64, int64 already.

‘frontal', ‘deploy’ are already encoded into nominal. dvcat', 'Survived', 'airbag', 'seatbelt', 'sex', 'abcat', 'occRole', 'injSeverity' are the categorical columns which should be encoded to nominal and are encoded to nominal by using data dictionary for few variables and by assigning a value to each category.

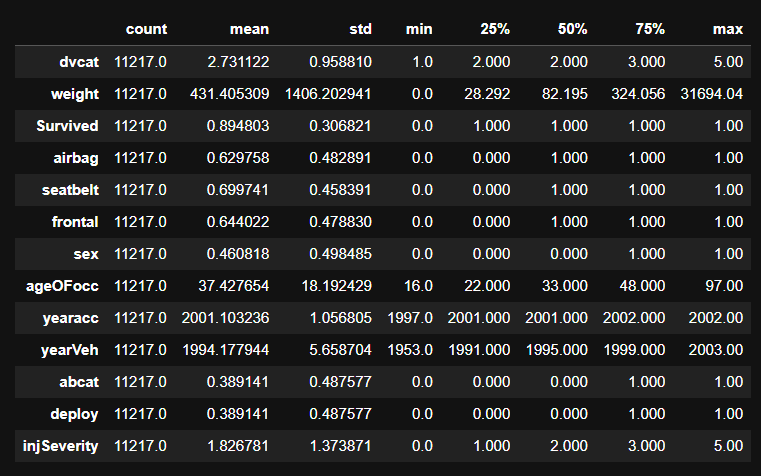
‘yearVeh’, ‘injSeverity’ are converted to type ‘int64’.

The top 5 observations after encoding are shown below.  
A picture containing screenshot

Description automatically generated

Fig2.1.6.2 Crash data sample observation after Encoding.

Data description is shown below:

  
Fig 2.1.6.3 Crash data statistical description after Encoding

At first look, Standard of living is high, Contraceptive used is Yes. Need to perform regression to understand more insights from the dataset.

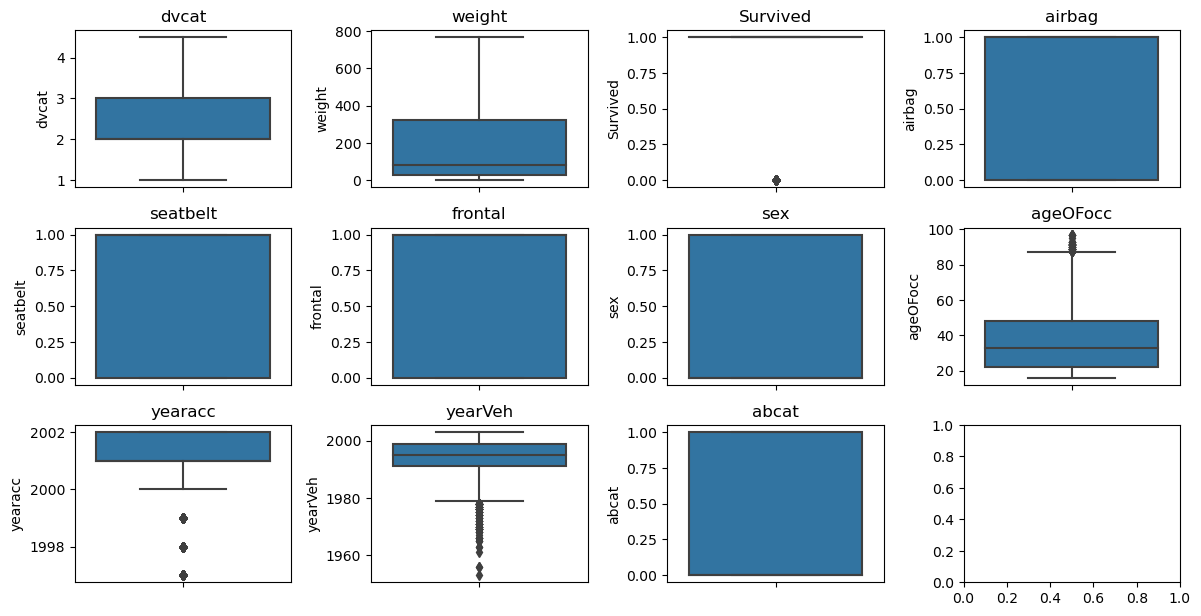
### Outliers

Below are the outliers after encoding the data.A picture containing diagram, screenshot, text, rectangle

Description automatically generated  
Fig2.1.7.1 Boxplot showing Crash Data outliers

Highest outliers are likely present in ‘weight’. The outliers in ‘ageOFocc’, ‘yearacc’, ‘yearVeh’ shall be ignored for now as these are very small number of outliers and also outliers shown in ‘Survived’ column are not outliers.

Outliers in the ‘dvcat’, ‘weight’ features are treated by adjusting them to the lower and upper bound values calculated by the IQR value. Below plot shows the final outliers.

  
Fig2.1.7.2 Boxplot showing Crash Data after treating outliers

## Prepare Data & Create Models

Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis). (8 marks)

Answer:

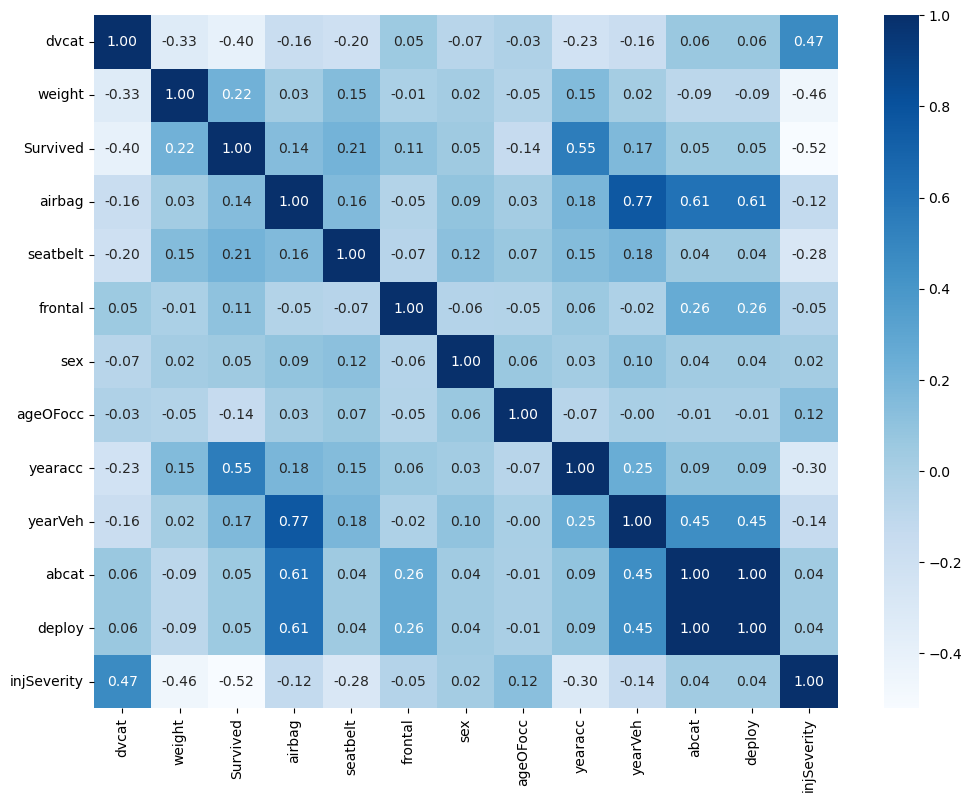
### Encode data

Encoding is performed in above section. The top 5 observations after encoding are shown below.  
A picture containing screenshot

Description automatically generated  
Fig 2.2.1.1 Crash data info after Encoding.

### Co-relation

The heatmap is plotted to show the correlation between variables.

  
Fig2.2.2.1 HeatMap showing Crash data correlation.

1. 'Survived' shows a negative correlation with 'dvcat' (-0.398861), indicating that higher levels of impact speed category are associated with a lower likelihood of survival.
2. 'Survived' has a positive correlation with 'weight' (0.217553), suggesting that heavier vehicles are more likely to have passengers who survived.
3. 'Survived' also has a positive correlation with 'yearacc' (0.549885), indicating that accidents in more recent years are associated with a higher likelihood of survival.
4. 'Survived' shows a positive correlation with 'abcat' (0.054346) and 'deploy' (0.054346), suggesting that the presence of an airbag and its deployment are associated with a slightly higher likelihood of survival.
5. 'Survived' exhibits a negative correlation with 'injSeverity' (-0.520610), implying that higher injury severity is associated with a lower likelihood of survival.
6. 'yearVeh' shows a relatively strong positive correlation with 'airbag' (0.766181) and 'deploy' (0.611983), indicating that newer vehicles are more likely to have airbags and their deployment.
7. 'weight' has a negative correlation with 'injSeverity' (-0.455644), implying that heavier vehicles are associated with lower injury severity.
8. 'injSeverity' shows a positive correlation with 'dvcat' (0.470624), indicating that higher levels of impact speed category are associated with higher injury severity.

These insights provide a glimpse into the relationships between different variables in the dataset. However, it's important to note that correlation does not imply causation, and further analysis and domain knowledge are necessary to draw meaningful conclusions.Top of Form

### Split Data

Using sklearn train test split method, the dataset is split into 70 30 for training and testing using random state 1 to replicate results, stratify on ‘Survived’ variable. This will ensure a balance in the data and will not cause biasness while Training or Testing the mode.

The sample train dataset is shown below:

A screenshot of a computer

Description automatically generated with low confidence  
Fig2.2.3.1 Crash data- Train Data sample

The sample test dataset is shown below:

A screenshot of a computer

Description automatically generated with low confidence

Fig2.2.3.2 Crash data- Test Data sample

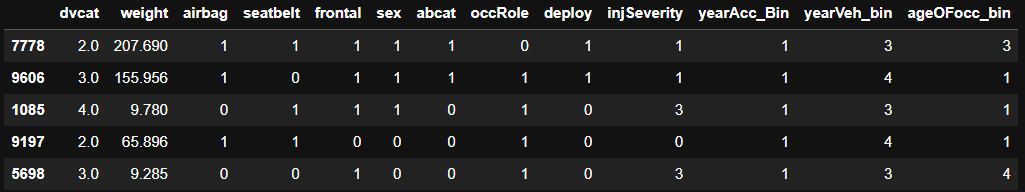


Fig2.2.3.3 Crash data train sample after binning

A screenshot of a computer

Description automatically generated with medium confidenceFig2.2.3.4 Crash data test sample after binning

However, note that the dataset has some degree of corelation & multi collinearity as shown in heatmaps in above section.

### Model 1 – Decision Tree

Decision tree model is built using criterion='gini', max\_depth=5, random\_state=0 hyperparameters. Decision trees model is as below:

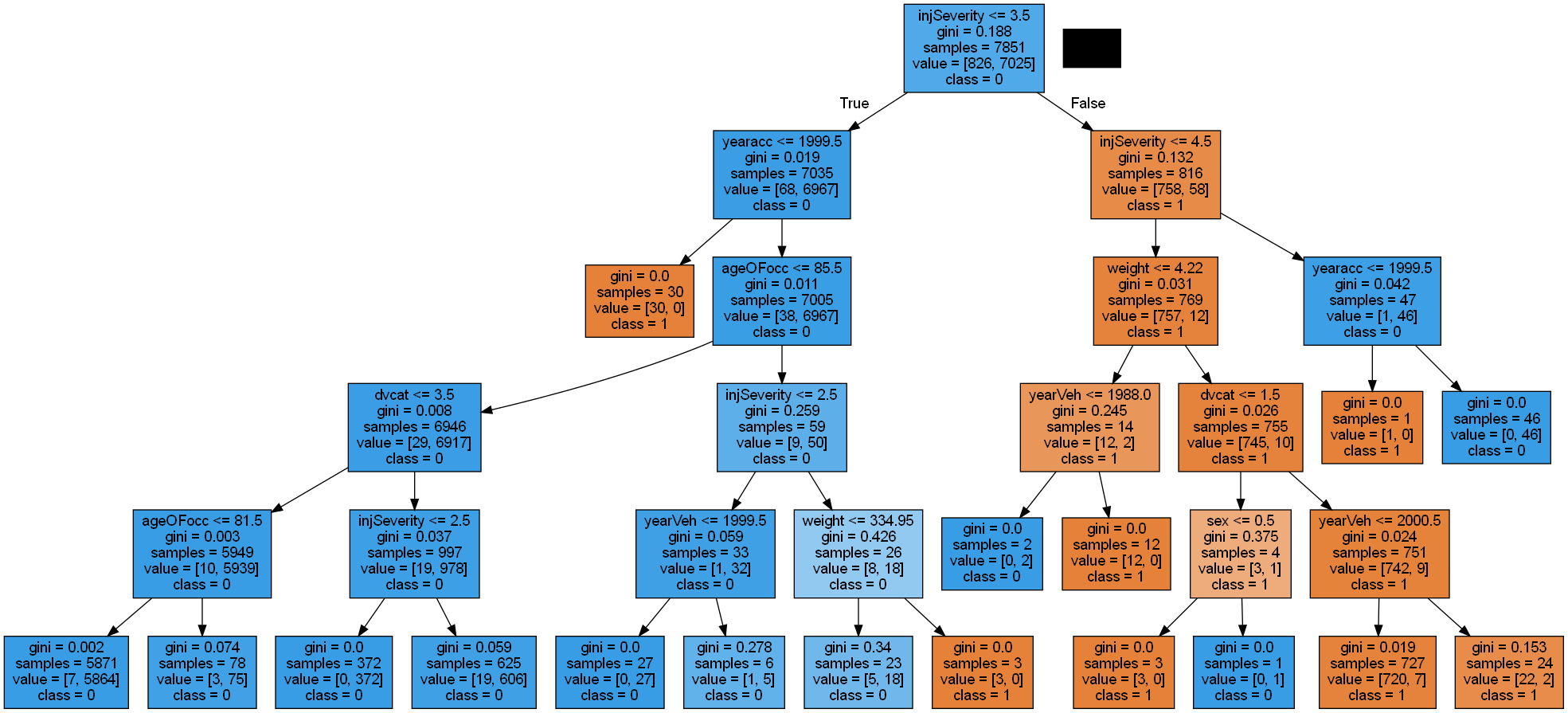
  
Fig2.2.4.1 Model -1 Decision Tree basic



Fig2.2.4.2 Importance of Features

The injSeverity, year of accident, weight, impact speed are the important features in determining the ‘Survived’ class.

### Model 2 – Apply Logistic, LDA, Cart

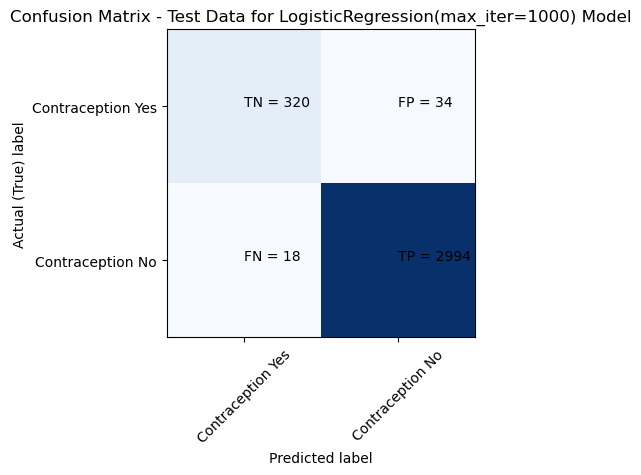
Logistic Regression, LDA, CART are applied on the dataset using training and performance is checked using the test data.

Hyper parameters used for the models are max\_iter=1000 in Logistic regression, n\_components=1 in LDA, criterion=’gini’ in CART.

### Model 3 – Apply Logistic, LDA, CART on binned data

A picture containing text, screenshot, line, plot

Description automatically generated



Model: Logistic Regression

Confusion Matrix for Train Data:

[[ 746 80]

[ 48 6977]]

Classification Report for Train Data:

precision recall f1-score support

0 0.94 0.90 0.92 826

1 0.99 0.99 0.99 7025

accuracy 0.98 7851

macro avg 0.96 0.95 0.96 7851

weighted avg 0.98 0.98 0.98 7851

-----------------------

Model: Logistic Regression

Confusion Matrix for Test Data:

[[ 320 34]

[ 18 2994]]

Classification Report for Test Data:

precision recall f1-score support

0 0.95 0.90 0.92 354

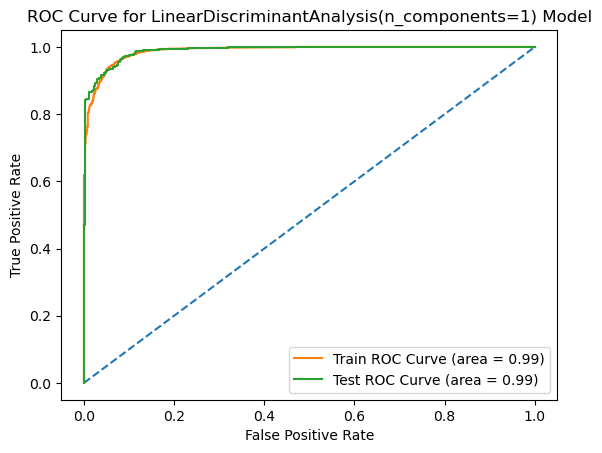
1 0.99 0.99 0.99 3012

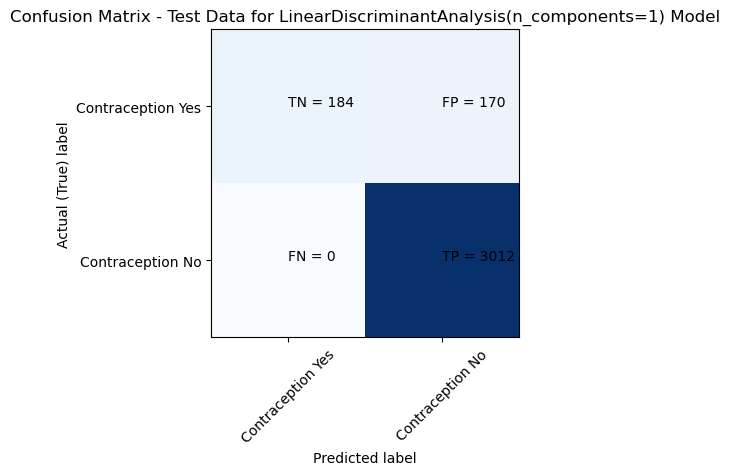
accuracy 0.98 3366

macro avg 0.97 0.95 0.96 3366

weighted avg 0.98 0.98 0.98 3366

-----------------------





Model: LDA

Confusion Matrix for Train Data:

[[ 438 388]

[ 0 7025]]

Classification Report for Train Data:

precision recall f1-score support

0 1.00 0.53 0.69 826

1 0.95 1.00 0.97 7025

accuracy 0.95 7851

macro avg 0.97 0.77 0.83 7851

weighted avg 0.95 0.95 0.94 7851

-----------------------

Model: LDA

Confusion Matrix for Test Data:

[[ 184 170]

[ 0 3012]]

Classification Report for Test Data:

precision recall f1-score support

0 1.00 0.52 0.68 354

1 0.95 1.00 0.97 3012

accuracy 0.95 3366

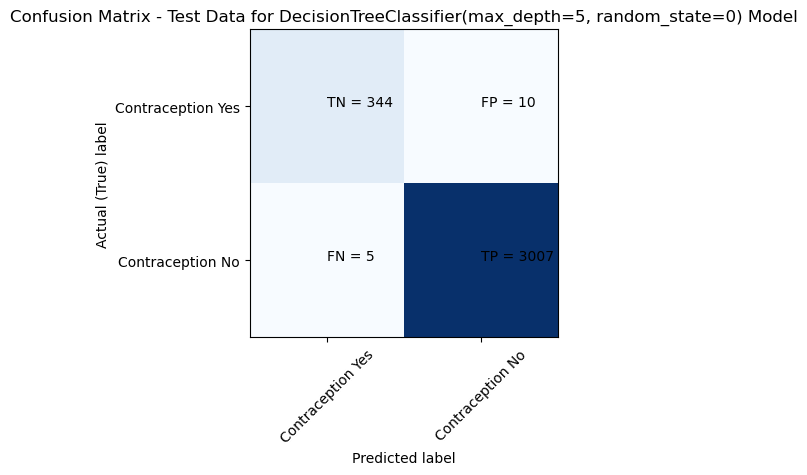
macro avg 0.97 0.76 0.83 3366

weighted avg 0.95 0.95 0.94 3366

-----------------------

A picture containing text, screenshot, line, plot

Description automatically generated



Model: CART

Confusion Matrix for Train Data:

[[ 788 38]

[ 9 7016]]

Classification Report for Train Data:

precision recall f1-score support

0 0.99 0.95 0.97 826

1 0.99 1.00 1.00 7025

accuracy 0.99 7851

macro avg 0.99 0.98 0.98 7851

weighted avg 0.99 0.99 0.99 7851

-----------------------

Model: CART

Confusion Matrix for Test Data:

[[ 344 10]

[ 5 3007]]

Classification Report for Test Data:

precision recall f1-score support

0 0.99 0.97 0.98 354

1 1.00 1.00 1.00 3012

accuracy 1.00 3366

macro avg 0.99 0.99 0.99 3366

weighted avg 1.00 1.00 1.00 3366

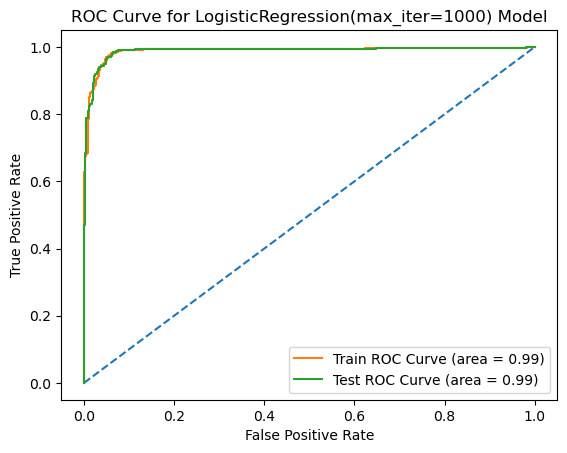
-----------------------

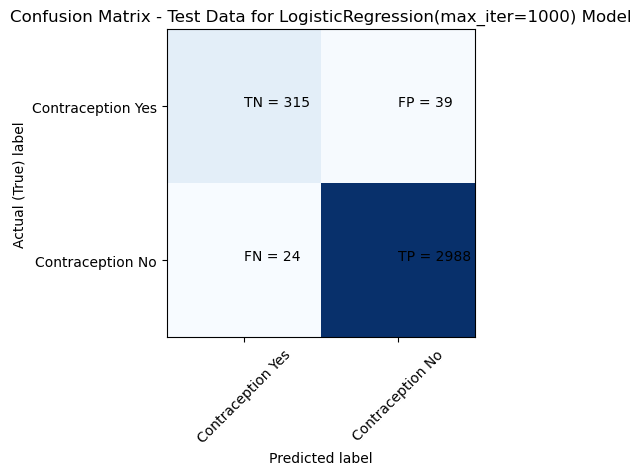
## Performance Metrics

Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Compare both the models and write inferences, which model is best/optimized. (8 marks)

Answer:

### ROC Curve, Classification Report, Confusion Matrix

When the model is run without any treatment of Feature Importances or multi -collinearity, below are the classification results. 



Model: Logistic Regression

Confusion Matrix for Train Data:

[[ 731 95]

[ 60 6965]]

Classification Report for Train Data:

precision recall f1-score support

0 0.92 0.88 0.90 826

1 0.99 0.99 0.99 7025

accuracy 0.98 7851

macro avg 0.96 0.94 0.95 7851

weighted avg 0.98 0.98 0.98 7851

-----------------------

Model: Logistic Regression

Confusion Matrix for Test Data:

[[ 315 39]

[ 24 2988]]

Classification Report for Test Data:

precision recall f1-score support

0 0.93 0.89 0.91 354

1 0.99 0.99 0.99 3012

accuracy 0.98 3366

macro avg 0.96 0.94 0.95 3366

weighted avg 0.98 0.98 0.98 3366

-----------------------

A picture containing text, screenshot, line, plot

Description automatically generated

A picture containing text, screenshot, diagram, font

Description automatically generated

Model: LDA

Confusion Matrix for Train Data:

[[ 565 261]

[ 50 6975]]

Classification Report for Train Data:

precision recall f1-score support

0 0.92 0.68 0.78 826

1 0.96 0.99 0.98 7025

accuracy 0.96 7851

macro avg 0.94 0.84 0.88 7851

weighted avg 0.96 0.96 0.96 7851

-----------------------

Model: LDA

Confusion Matrix for Test Data:

[[ 247 107]

[ 28 2984]]

Classification Report for Test Data:

precision recall f1-score support

0 0.90 0.70 0.79 354

1 0.97 0.99 0.98 3012

accuracy 0.96 3366

macro avg 0.93 0.84 0.88 3366

weighted avg 0.96 0.96 0.96 3366

-----------------------

A picture containing text, screenshot, line, plot

Description automatically generated

A picture containing text, screenshot, diagram, font

Description automatically generated

Model: CART

Confusion Matrix for Train Data:

[[ 791 35]

[ 9 7016]]

Classification Report for Train Data:

precision recall f1-score support

0 0.99 0.96 0.97 826

1 1.00 1.00 1.00 7025

accuracy 0.99 7851

macro avg 0.99 0.98 0.98 7851

weighted avg 0.99 0.99 0.99 7851

-----------------------

Model: CART

Confusion Matrix for Test Data:

[[ 337 17]

[ 5 3007]]

Classification Report for Test Data:

precision recall f1-score support

0 0.99 0.95 0.97 354

1 0.99 1.00 1.00 3012

accuracy 0.99 3366

macro avg 0.99 0.98 0.98 3366

weighted avg 0.99 0.99 0.99 3366

-----------------------

### Accuracy & ROC

A screenshot of a graph

Description automatically generated with low confidence

Fig2.3.2 Crash Data Regression Performance Metrics – Accuracy & ROC

CART is the best model here when VIF is not treated.

## Inference

Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Compare both the models and write inferences, which model is best/optimized. (8 marks)

Answer:

### Model Discussion

Considering the importance of this classification about ‘survive’, a model that has best recall and low FP is selected i.e. CART.

1. Both Logistic Regression and CART models perform well in terms of F1-score and accuracy across both iterations. They have F1-scores above 0.90 and high accuracies for both the train and test data.
2. LDA, on the other hand, shows lower F1-scores and accuracies compared to the other models
3. CART consistently give better results than the other models in terms of F1-score and accuracy.
4. Logistic Regression performs slightly better than LDA, achieving higher F1-scores and accuracies.
5. Considering the high performance of CART and Logistic Regression, they can be considered as strong models for predicting the classification task.
6. When multi-collinearity is treated, we shall get a even cleaner and simpler decision tree.

### Business Recommendations:

1. The injSeverity, year of accident, weight, impact speed are the important features in determining the ‘Survived’ class. Conduct safety awareness programs and monitor ‘airbags’ or similar safety features.
2. Age and car model did not actor is as much contrary to popular belief. We can suggest hence to keep collecting data to understand hotspots, trends. And keep updating the safety policies. As, regular evaluation and updates to road safety policies based on emerging research.

# Reflection Report:

Please reflect on all that you learnt and fill this reflection report. You have to copy the link and paste it on the URL bar of your respective browser. https://docs.google.com/forms/d/e/1FAIpQLScKuVyrmTTM7Pboh0IB4YIBUbJp2NrDZcsY4SCRn3ZUkwmLGg/viewform

<Completed>