# PREDICTIVE MODELLING

NAME: **ANISH DASGUPTA**

COURSE: **DSBA**

BATCH: **May'21**

SUBJECT**: Alternate Project for Predictive Modelling**

# TABLE OF CONTENTS:

# DATA DICTIONARY:

| Term | Description |
|---|---|
| **Firm_level_data** | |
| sales | Sales (in millions of dollars). |
| capital | Net stock of property, plant, and equipment. |
| patents | Granted patents. |
| randd | R&D stock (in millions of dollars). |
| employment | Employment (in 1000s). |

| | |
|---|---|
| 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 |
| 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. |
| value | Stock market value. |
| institutions | Proportion of stock owned by institutions. |
| **Car_Crash** | |
| dvcat | factor with levels (estimated impact speeds) 1-9km/h, 10-24, 25-39, 40-54, 55+ |
| 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) |
| Survived | factor with levels Survived or not_survived |
| airbag | a factor with levels none or airbag |
| seatbelt | a factor with levels none or belted |
| frontal | a numeric vector; 0 = non-frontal, 1=frontal impact |
| sex | a factor with levels f: Female or m: Male |
| ageOFocc | age of occupant in years |
| yearacc | year of accident |
| yearVeh | Year of model of vehicle; a numeric vector |
| abcat | Did one or more (driver or passenger) airbag(s) deploy? This factor has levels deploy, nodeploy and unavail |
| occRole | a factor with levels driver or pass: passenger |
| deploy | a numeric vector: 0 if an airbag was unavailable or did not deploy; 1 if one or more bags deployed |
| injSeverity | a numeric vector; 0: none, 1: possible injury, 2: no incapacity, 3: incapacity, 4: killed; 5: unknown, 6: prior death |
| 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. |

# OBJECTIVE:

The primary objective of this report is to provide the business implications of the presented problem statements. The insights provided in this report primarily analyze the problem under hand and attempts to answer the questions that follow it. The codes for deriving those insights are maintained separately.

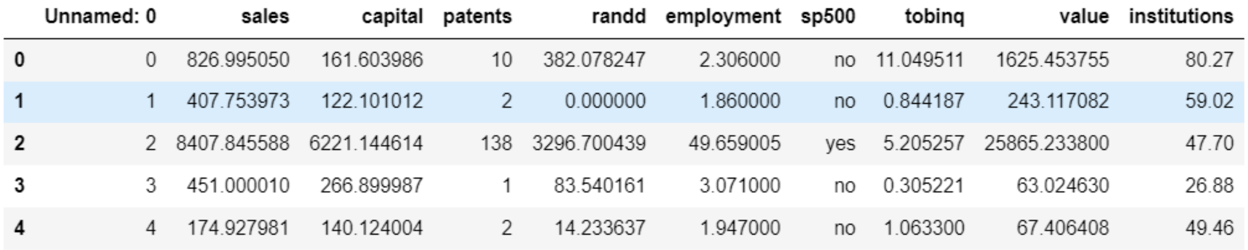
# Problem Statement 1: Linear Regression

You are a part of an investment firm and your work is to do research about these 759 firms. You are provided with the dataset containing the sales and other attributes of these 759 firms. 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.

## 1.1) 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.

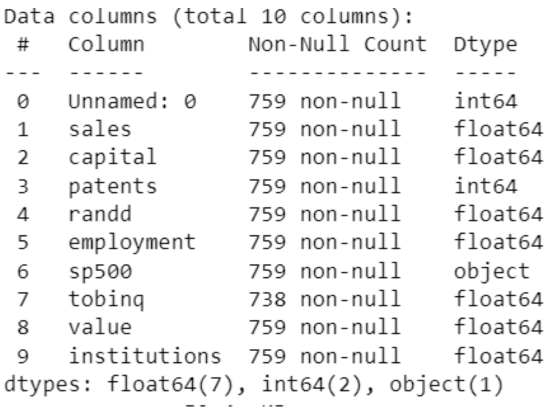
### Data Report:

**Raw Data -**



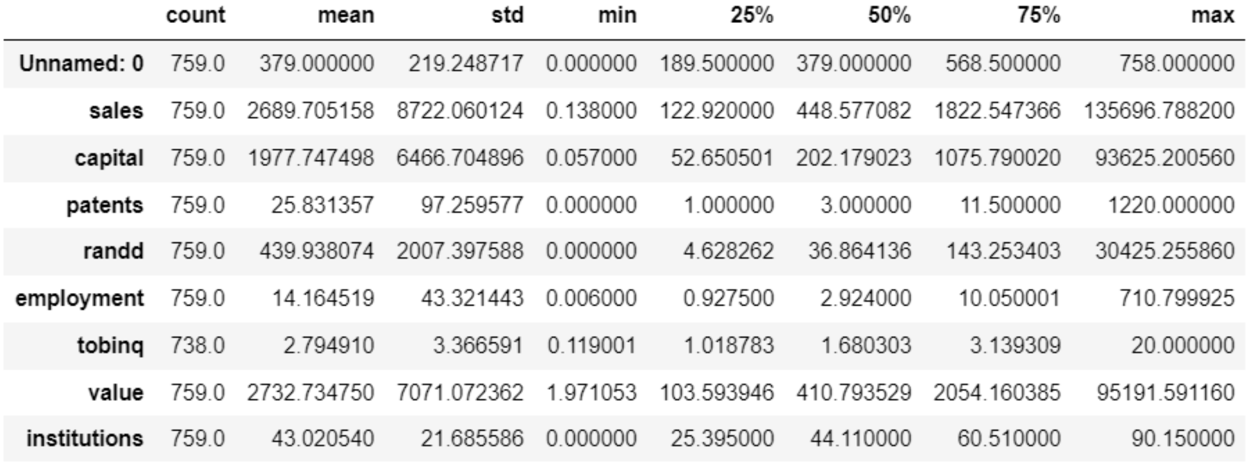
| | Unnamed: 0 | sales | capital | patents | randd | employment | sp500 | tobinq | value | institutions |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 826.995050 | 161.603986 | 10 | 382.078247 | 2.306000 | no | 11.049511 | 1625.453755 | 80.27 |
| **1** | 1 | 407.753973 | 122.101012 | 2 | 0.000000 | 1.860000 | no | 0.844187 | 243.117082 | 59.02 |
| **2** | 2 | 8407.845588 | 6221.144614 | 138 | 3296.700439 | 49.659005 | yes | 5.205257 | 25865.233800 | 47.70 |
| **3** | 3 | 451.000010 | 266.899987 | 1 | 83.540161 | 3.071000 | no | 0.305221 | 63.024630 | 26.88 |
| **4** | 4 | 174.927981 | 140.124004 | 2 | 14.233637 | 1.947000 | no | 1.063300 | 67.406408 | 49.46 |

**Data Info -**



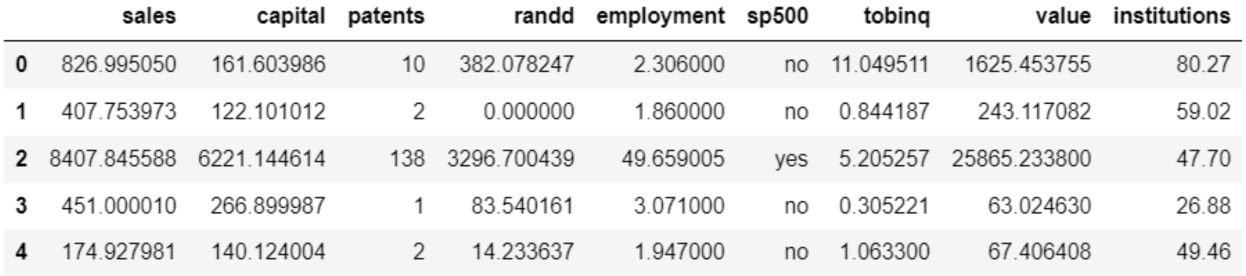
```
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    759 non-null    int64  
 1   sales         759 non-null    float64
 2   capital       759 non-null    float64
 3   patents       759 non-null    int64  
 4   randd         759 non-null    float64
 5   employment    759 non-null    float64
 6   sp500         759 non-null    object 
 7   tobinq        738 non-null    float64
 8   value         759 non-null    float64
 9   institutions  759 non-null    float64
dtypes: float64(7), int64(2), object(1)
```

**Data Description –**



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Unnamed: 0** | 759.0 | 379.000000 | 219.248717 | 0.000000 | 189.500000 | 379.000000 | 568.500000 | 758.000000 |
| **sales** | 759.0 | 2689.705158 | 8722.060124 | 0.138000 | 122.920000 | 448.577082 | 1822.547366 | 135696.788200 |
| **capital** | 759.0 | 1977.747498 | 6466.704896 | 0.057000 | 52.650501 | 202.179023 | 1075.790020 | 93625.200560 |
| **patents** | 759.0 | 25.831357 | 97.259577 | 0.000000 | 1.000000 | 3.000000 | 11.500000 | 1220.000000 |
| **randd** | 759.0 | 439.938074 | 2007.397588 | 0.000000 | 4.628262 | 36.864136 | 143.253403 | 30425.255860 |
| **employment** | 759.0 | 14.164519 | 43.321443 | 0.006000 | 0.927500 | 2.924000 | 10.050001 | 710.799925 |
| **tobinq** | 738.0 | 2.794910 | 3.366591 | 0.119001 | 1.018783 | 1.680303 | 3.139309 | 20.000000 |
| **value** | 759.0 | 2732.734750 | 7071.072362 | 1.971053 | 103.593946 | 410.793529 | 2054.160385 | 95191.591160 |
| **institutions** | 759.0 | 43.020540 | 21.685586 | 0.000000 | 25.395000 | 44.110000 | 60.510000 | 90.150000 |

## Dropping Unique Identifier –



| | sales | capital | patents | randd | employment | sp500 | tobinq | value | institutions |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 826.995050 | 161.603986 | 10 | 382.078247 | 2.306000 | no | 11.049511 | 1625.453755 | 80.27 |
| **1** | 407.753973 | 122.101012 | 2 | 0.000000 | 1.860000 | no | 0.844187 | 243.117082 | 59.02 |
| **2** | 8407.845588 | 6221.144614 | 138 | 3296.700439 | 49.659005 | yes | 5.205257 | 25865.233800 | 47.70 |
| **3** | 451.000010 | 266.899987 | 1 | 83.540161 | 3.071000 | no | 0.305221 | 63.024630 | 26.88 |
| **4** | 174.927981 | 140.124004 | 2 | 14.233637 | 1.947000 | no | 1.063300 | 67.406408 | 49.46 |

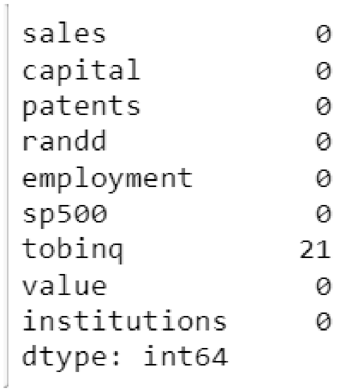
**Insights from Numerical fields:**

1. We have sales, capital, patents, randd, employment, tobinq, value and institutions as numerical fields.
2. We need to check if the data for these numerical fields are not below zero since these values are not meant to be negative.
3. We also need to make sure that the field 'institutions' does not have any values greater than 100 since it's a proportion field.
4. We also need to make sure that the field 'employment' does not have any values less than .001(thousands), since the number of employees cannot be less than 1.
5. We see that none of the numerical fields have any negative values.
6. We also see that the 'employment' and 'institutions' also have values as per the constraints.

**Insights from Categorical fields:**

We see that the categorical field does not have any values other than the presented 'yes' or 'no'.

## Null Values –

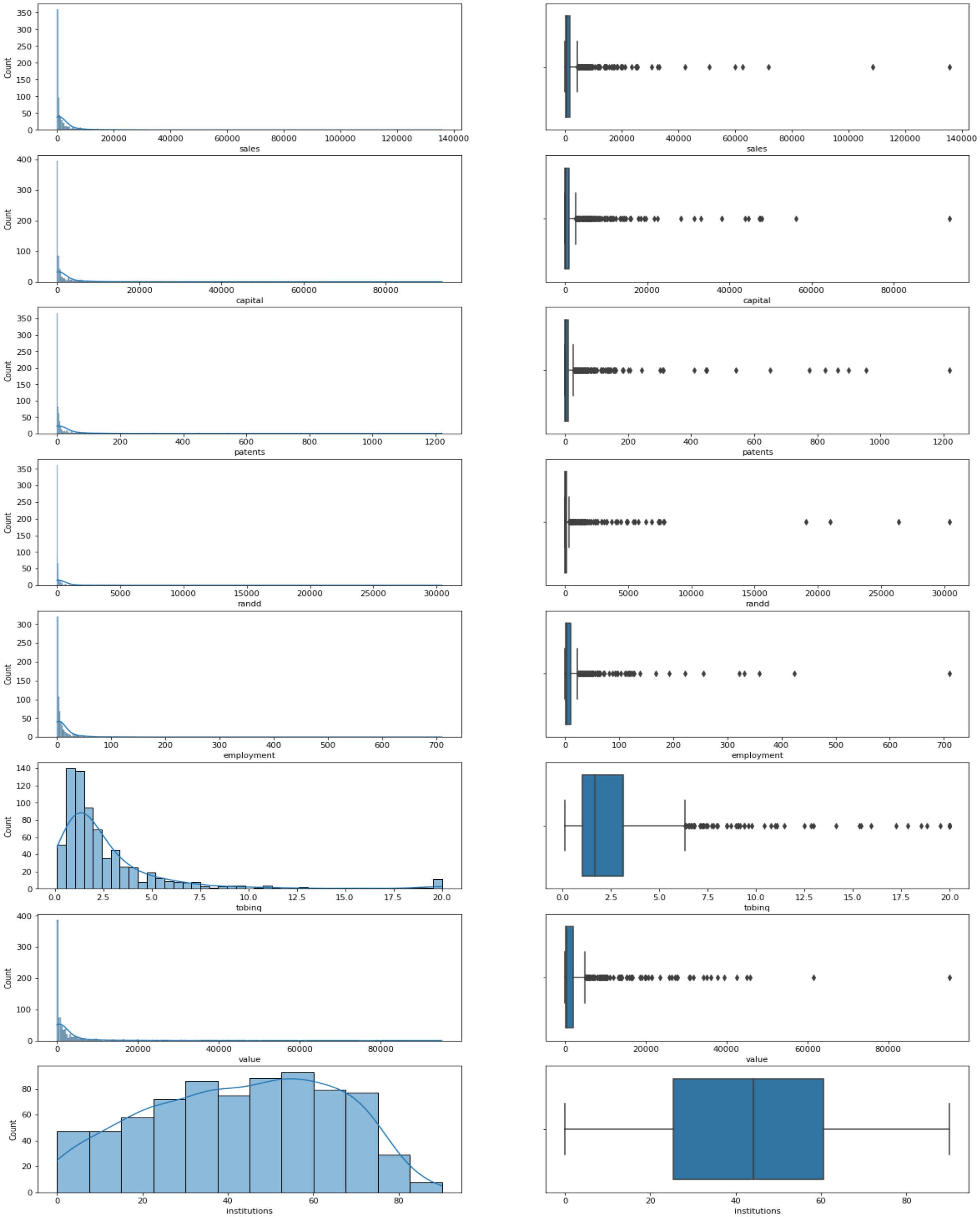


```
sales            0
capital          0
patents          0
randd            0
employment       0
sp500            0
tobinq          21
value            0
institutions     0
dtype: int64
```

**Insights:**

1. We see that the data set has 759 data points and 9 predictors.
2. We see that the 'tobinq' field has few null values which will need ot be treated. Since the number of null values are very less compared to the total data points, we can impute these values.
3. We also observe that there are no duplicate values.

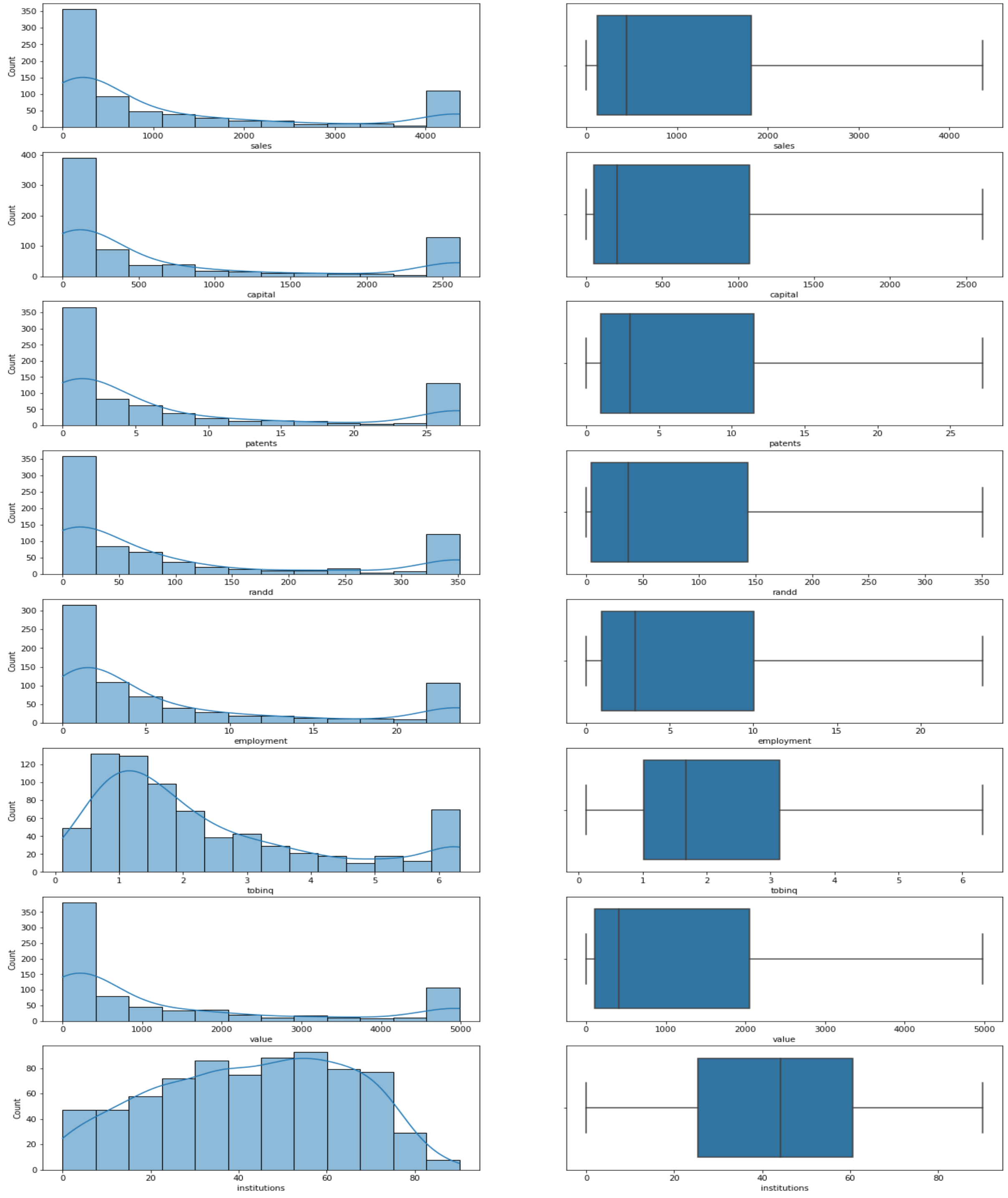
# Univariate Analysis:



## Insights:

We can see that there are a lot of outliers present for the numerical fields resulting in the extreme skewness of the plots. Hence we need to go for outlier treatment.

## After the Outlier treatment –

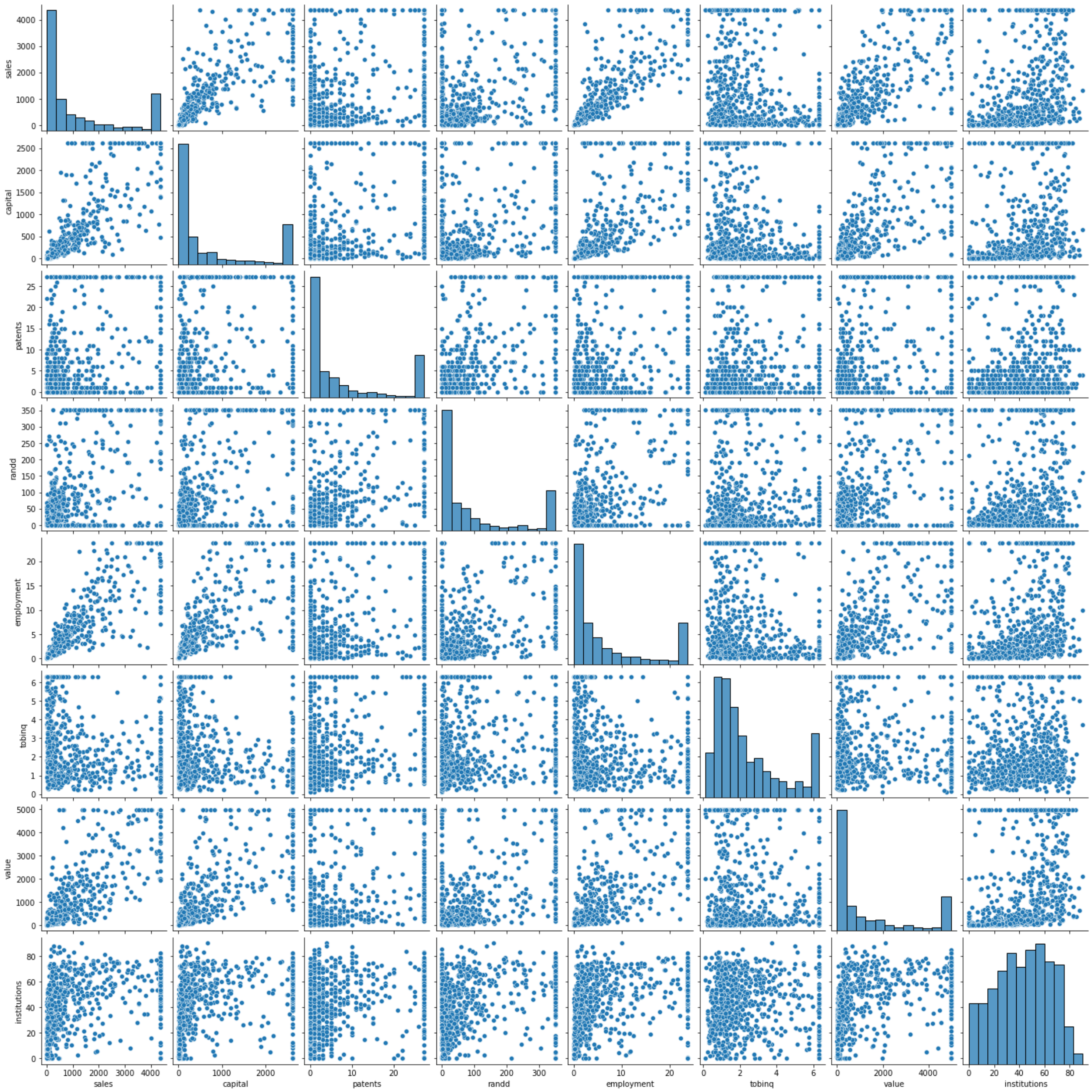


We can see that the outliers have been treated successfully.

**Insights from Univariate Analysis:**

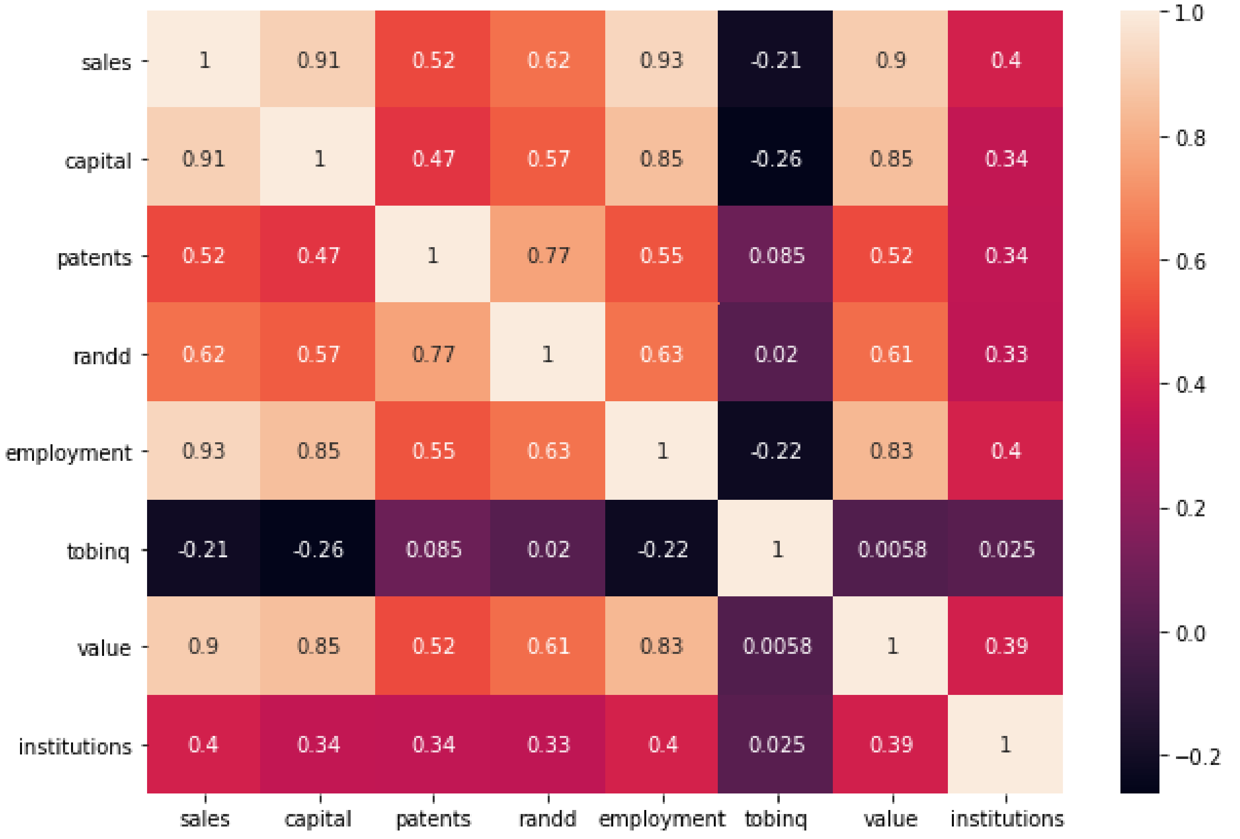
1. From the 'sales' plot, we can see that more than 500 companies have sales less than or equal to 1000 million dollars.
2. From the 'capital' plot, we see that majority(<75%) of the companies have capital less than 500.
3. From the 'patents' plot, we observe that majority of the companies have 0 to 5 patents.
4. From the 'tobinq' plot, we can see that most of the companies have the values from 0.5 to 2 tobinq ratio value.
5. All the fields (except 'institutions') are heavily right skewed. This means that there are less number of companies which are doing exceptionally well in terms of sales, capital and working efficiency.

# Bivariate Analysis:



**Insights from Pairplot:**

1. We see a decent positive correlation pattern between sales and capital. This is because of the fact that in order to generate high sales, the company needs to maintain high capital as well.
2. We see a strong positive correlation pattern between sales and employment, since high sales imply that the company is profitable and comfortable hiring high number of employees. Inversely, the higher number of employees, they help generate more sales.
3. We see a decent positive correlation pattern between sales and value, indicating that high sales generating companies are valued higher in terms of stock prices.

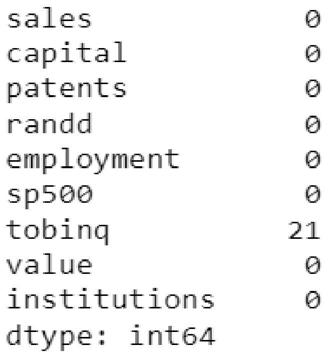


**Insights from Heatmap:**

1. We can observe similar case where sales is highly positively correlated with capital, value, employment, randd.
2. We can observe that sales has decent correlation with the number of patents the company holds as well as randd, since good RD helps in high sales for the company.
3. We also see that randd is heavily positively correlated with number of patents company holds.
4. We also observe that the tobinq value is slightly negatively correlated with sales, meaning higher the sales, lower the tobin1 value.

## 1.2) Impute null values if present? Do you think scaling is necessary in this case?

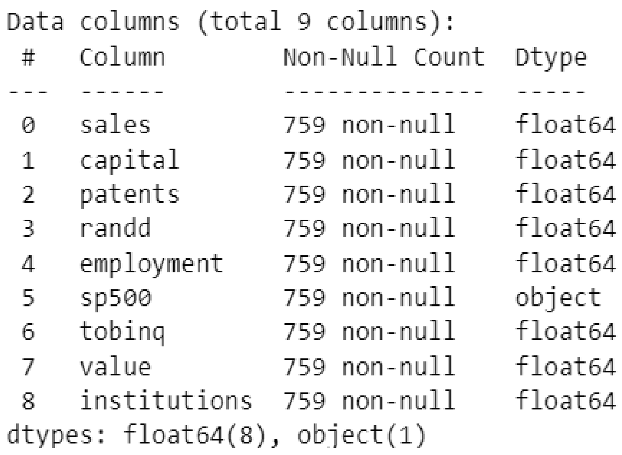
### Null Values:



```
sales           0
capital         0
patents         0
randd           0
employment      0
sp500           0
tobinq         21
value           0
institutions    0
dtype: int64
```

We see that the field 'tobinq' has 21 data points with null values, which need to be imputed.

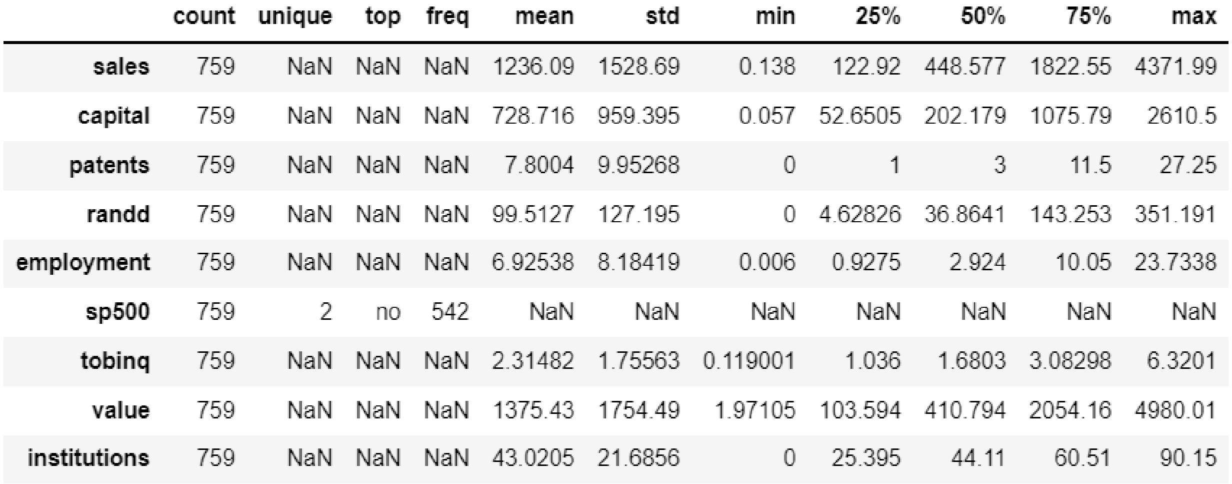
**After Null value treatment –**



```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sales         759 non-null    float64
 1   capital       759 non-null    float64
 2   patents       759 non-null    float64
 3   randd         759 non-null    float64
 4   employment    759 non-null    float64
 5   sp500         759 non-null    object 
 6   tobinq        759 non-null    float64
 7   value         759 non-null    float64
 8   institutions  759 non-null    float64
dtypes: float64(8), object(1)
```

We see that all the null values have been treated and imputed with the median value of the field. We have not used mean because of presence of outliers.

## Scaling:

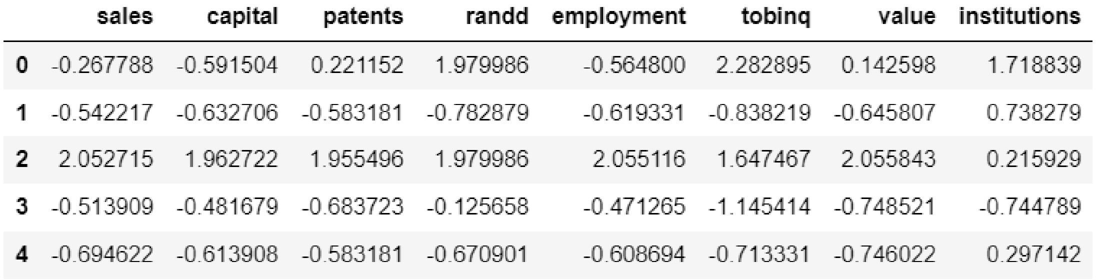


| | count | unique | top | freq | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **sales** | 759 | NaN | NaN | NaN | 1236.09 | 1528.69 | 0.138 | 122.92 | 448.577 | 1822.55 | 4371.99 |
| **capital** | 759 | NaN | NaN | NaN | 728.716 | 959.395 | 0.057 | 52.6505 | 202.179 | 1075.79 | 2610.5 |
| **patents** | 759 | NaN | NaN | NaN | 7.8004 | 9.95268 | 0 | 1 | 3 | 11.5 | 27.25 |
| **randd** | 759 | NaN | NaN | NaN | 99.5127 | 127.195 | 0 | 4.62826 | 36.8641 | 143.253 | 351.191 |
| **employment** | 759 | NaN | NaN | NaN | 6.92538 | 8.18419 | 0.006 | 0.9275 | 2.924 | 10.05 | 23.7338 |
| **sp500** | 759 | 2 | no | 542 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **tobinq** | 759 | NaN | NaN | NaN | 2.31482 | 1.75563 | 0.119001 | 1.036 | 1.6803 | 3.08298 | 6.3201 |
| **value** | 759 | NaN | NaN | NaN | 1375.43 | 1754.49 | 1.97105 | 103.594 | 410.794 | 2054.16 | 4980.01 |
| **institutions** | 759 | NaN | NaN | NaN | 43.0205 | 21.6856 | 0 | 25.395 | 44.11 | 60.51 | 90.15 |

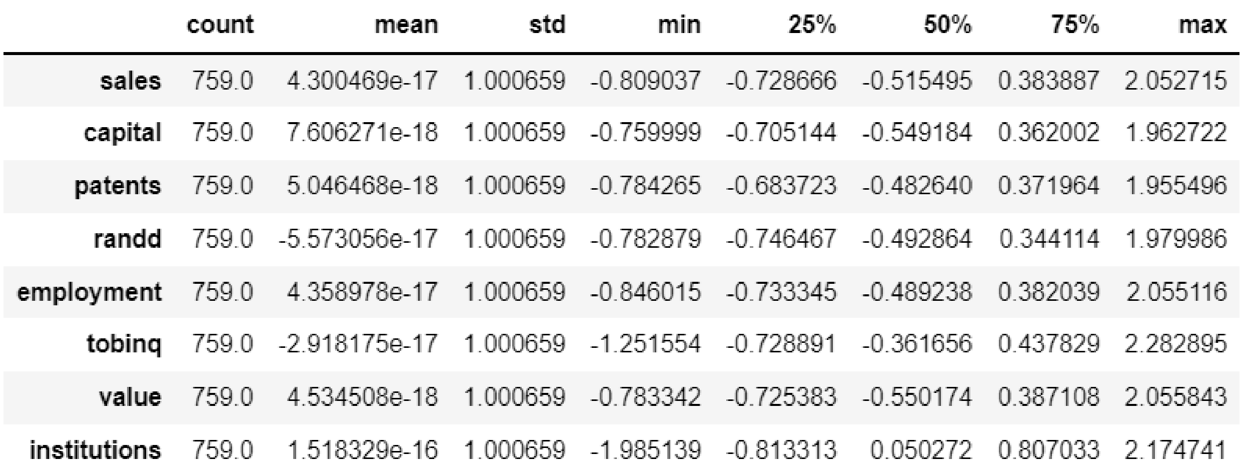
From the above data, we see that different fields have different scales of data. Hence we need to scale the data to cure the data set from being influenced by heavy weightages of field with respect to the data it stores.

Scaling the data in this case becomes necessary since the linear regression model might be biased with larger value data points.

## After Scaling of numerical fields –



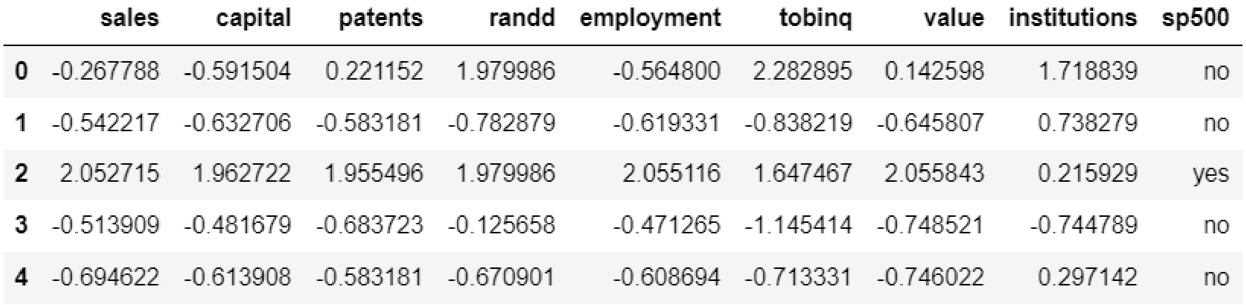
| | sales | capital | patents | randd | employment | tobinq | value | institutions |
|---|---|---|---|---|---|---|---|---|
| **0** | -0.267788 | -0.591504 | 0.221152 | 1.979986 | -0.564800 | 2.282895 | 0.142598 | 1.718839 |
| **1** | -0.542217 | -0.632706 | -0.583181 | -0.782879 | -0.619331 | -0.838219 | -0.645807 | 0.738279 |
| **2** | 2.052715 | 1.962722 | 1.955496 | 1.979986 | 2.055116 | 1.647467 | 2.055843 | 0.215929 |
| **3** | -0.513909 | -0.481679 | -0.683723 | -0.125658 | -0.471265 | -1.145414 | -0.748521 | -0.744789 |
| **4** | -0.694622 | -0.613908 | -0.583181 | -0.670901 | -0.608694 | -0.713331 | -0.746022 | 0.297142 |



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **sales** | 759.0 | 4.300469e-17 | 1.000659 | -0.809037 | -0.728666 | -0.515495 | 0.383887 | 2.052715 |
| **capital** | 759.0 | 7.606271e-18 | 1.000659 | -0.759999 | -0.705144 | -0.549184 | 0.362002 | 1.962722 |
| **patents** | 759.0 | 5.046468e-18 | 1.000659 | -0.784265 | -0.683723 | -0.482640 | 0.371964 | 1.955496 |
| **randd** | 759.0 | -5.573056e-17 | 1.000659 | -0.782879 | -0.746467 | -0.492864 | 0.344114 | 1.979986 |
| **employment** | 759.0 | 4.358978e-17 | 1.000659 | -0.846015 | -0.733345 | -0.489238 | 0.382039 | 2.055116 |
| **tobinq** | 759.0 | -2.918175e-17 | 1.000659 | -1.251554 | -0.728891 | -0.361656 | 0.437829 | 2.282895 |
| **value** | 759.0 | 4.534508e-18 | 1.000659 | -0.783342 | -0.725383 | -0.550174 | 0.387108 | 2.055843 |
| **institutions** | 759.0 | 1.518329e-16 | 1.000659 | -1.985139 | -0.813313 | 0.050272 | 0.807033 | 2.174741 |

We can see that the numerical part of the data set has been scaled successfully as the standard deviation of all the fields ~ 1 and mean of all the fields ~ 0

## Final Scaled Data (after concatenating categorical fields) –



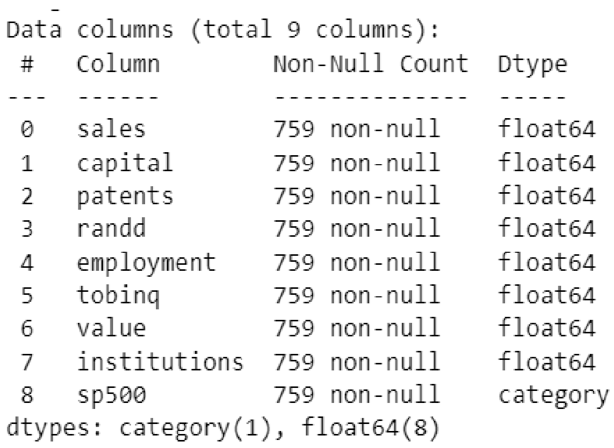
| | sales | capital | patents | randd | employment | tobinq | value | institutions | sp500 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | -0.267788 | -0.591504 | 0.221152 | 1.979986 | -0.564800 | 2.282895 | 0.142598 | 1.718839 | no |
| **1** | -0.542217 | -0.632706 | -0.583181 | -0.782879 | -0.619331 | -0.838219 | -0.645807 | 0.738279 | no |
| **2** | 2.052715 | 1.962722 | 1.955496 | 1.979986 | 2.055116 | 1.647467 | 2.055843 | 0.215929 | yes |
| **3** | -0.513909 | -0.481679 | -0.683723 | -0.125658 | -0.471265 | -1.145414 | -0.748521 | -0.744789 | no |
| **4** | -0.694622 | -0.613908 | -0.583181 | -0.670901 | -0.608694 | -0.713331 | -0.746022 | 0.297142 | no |

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

Encoding is necessary since python does not understand string values while applying modelling algorithms.

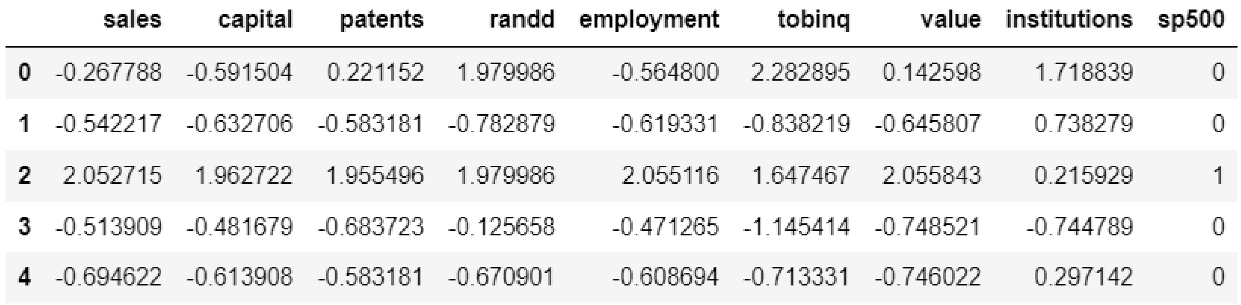
We see that the field sp500 has only 2 unique values i.e 'yes' or 'no'. Hence we can go ahead with One hot encoding since the number of the categories is very less and the categories are nominal and not ordinal in nature.

### Converting the categorical object field to category type –

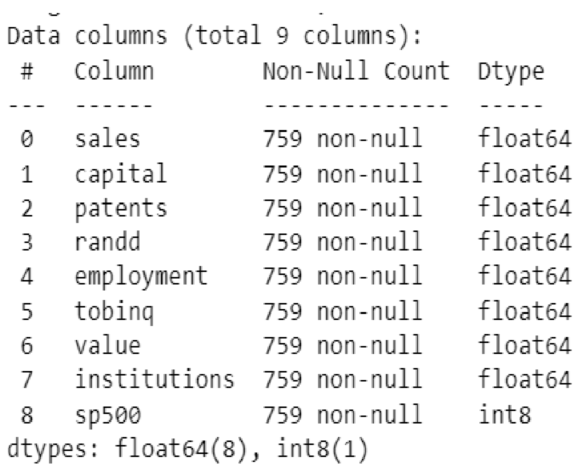


```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sales         759 non-null    float64
 1   capital       759 non-null    float64
 2   patents       759 non-null    float64
 3   randd         759 non-null    float64
 4   employment    759 non-null    float64
 5   tobinq        759 non-null    float64
 6   value         759 non-null    float64
 7   institutions  759 non-null    float64
 8   sp500         759 non-null    category
dtypes: category(1), float64(8)
```

### After One Hot encoding –



| | sales | capital | patents | randd | employment | tobinq | value | institutions | sp500 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | -0.267788 | -0.591504 | 0.221152 | 1.979986 | -0.564800 | 2.282895 | 0.142598 | 1.718839 | 0 |
| **1** | -0.542217 | -0.632706 | -0.583181 | -0.782879 | -0.619331 | -0.838219 | -0.645807 | 0.738279 | 0 |
| **2** | 2.052715 | 1.962722 | 1.955496 | 1.979986 | 2.055116 | 1.647467 | 2.055843 | 0.215929 | 1 |
| **3** | -0.513909 | -0.481679 | -0.683723 | -0.125658 | -0.471265 | -1.145414 | -0.748521 | -0.744789 | 0 |
| **4** | -0.694622 | -0.613908 | -0.583181 | -0.670901 | -0.608694 | -0.713331 | -0.746022 | 0.297142 | 0 |



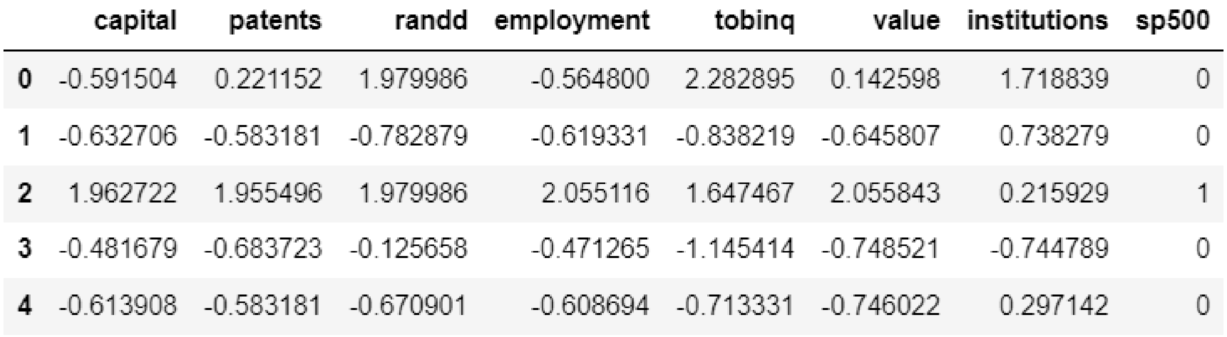
```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sales         759 non-null    float64
 1   capital       759 non-null    float64
 2   patents       759 non-null    float64
 3   randd         759 non-null    float64
 4   employment    759 non-null    float64
 5   tobinq        759 non-null    float64
 6   value         759 non-null    float64
 7   institutions  759 non-null    float64
 8   sp500         759 non-null    int8   
dtypes: float64(8), int8(1)
```

We can see that the encoding of the categorical field has been completed successfully. The data is now ready for modelling.

## Modelling:

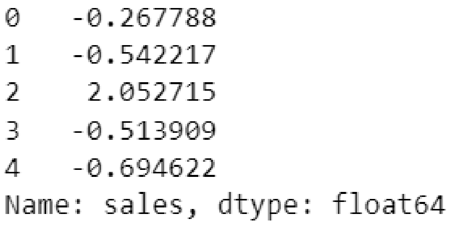
We know that our target variable is 'sales'. Hence we need to divide the data into target and predictors.

### Predictor variable set-



| | capital | patents | randd | employment | tobinq | value | institutions | sp500 |
|---|---|---|---|---|---|---|---|---|
| **0** | -0.591504 | 0.221152 | 1.979986 | -0.564800 | 2.282895 | 0.142598 | 1.718839 | 0 |
| **1** | -0.632706 | -0.583181 | -0.782879 | -0.619331 | -0.838219 | -0.645807 | 0.738279 | 0 |
| **2** | 1.962722 | 1.955496 | 1.979986 | 2.055116 | 1.647467 | 2.055843 | 0.215929 | 1 |
| **3** | -0.481679 | -0.683723 | -0.125658 | -0.471265 | -1.145414 | -0.748521 | -0.744789 | 0 |
| **4** | -0.613908 | -0.583181 | -0.670901 | -0.608694 | -0.713331 | -0.746022 | 0.297142 | 0 |

### Target variable set-



```
0   -0.267788
1   -0.542217
2    2.052715
3   -0.513909
4   -0.694622
Name: sales, dtype: float64
```

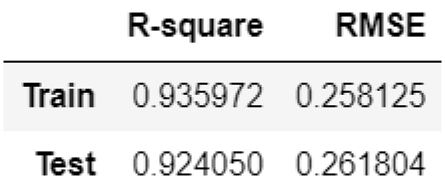
For the application of Linear regression model on our data, we split our data into 70:30 ratio where the training data set will comprise of the 70% of the data set and testing data set will comprise of the rest 30%.

The linear regression model will be trained on the training data set, where the model will pick up the statistics of the train data and learn the prediction. The trained model will be then tested on the test data set to see how the model performs.

## Performance Metrics:

For linear models, we have 2 major performance metrics to understand if the model performed sufficiently, namely, R-squared and RMSE(Root Mean Squared Error).

We check both these metrics for our train data as well as test data.



| | R-square | RMSE |
|---|---|---|
| **Train** | 0.935972 | 0.258125 |
| **Test** | 0.924050 | 0.261804 |

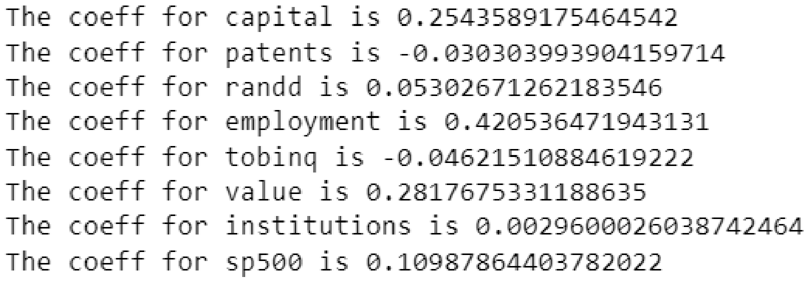
**Insights:**

1. For our predicted model, we see that the RMSE values for the train and the test set is extremely close, which inferences that the model has good performance and is neither under or over fit.

2. We can also observe that the R-square value values for both train and test are very close, indicating that the trained model is explainable for both the data sets.

## 1.4) Inference: Based on these predictions, what are the business insights and recommendations.

We check the feature importance which gives us an understanding of which features factor the most while generating the predictive model. This metric is generated by the gradient of the field in question. Higher the gradient, higher is the influence of the attribute while model building.



```
The coeff for capital is 0.2543589175464542
The coeff for patents is -0.030303993904159714
The coeff for randd is 0.05302671262183546
The coeff for employment is 0.420536471943131
The coeff for tobinq is -0.04621510884619222
The coeff for value is 0.2817675331188635
The coeff for institutions is 0.0029600026038742464
The coeff for sp500 is 0.10987864403782022
```

### Business Insights:

1. Since we are working for an investment firm, we need to look at the firm data with a view of generating insights so that our firm's investment is safe and profitable.

2. We see that the top 5 most important and influencing predictors for the sales are:

i) employment
ii) value
iii) capital
iv) sp500
v) randd

This helps us to understand which predictors need to be given priorities while choosing the firms for an investment opportunity.

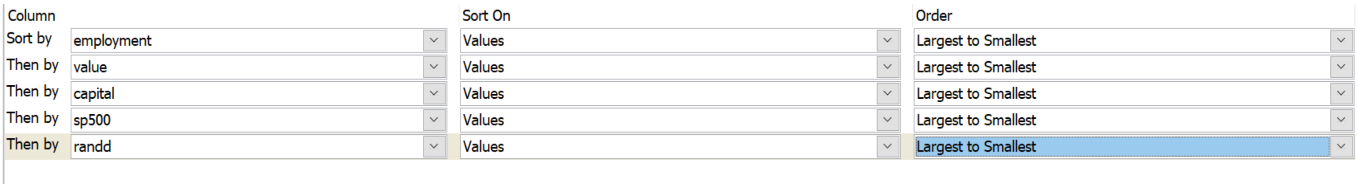
3. By building a predictive model, we see that our model has an accuracy(r-square) of ~92%. The firms which perform better in sales than our predicted sales are good options while being considered for an investment.

4. We successfully decreased the universe of the firms by ~ 50% by choosing firms which did better than the predicted model values and we got **383** firms to choose from finally.

5. From the curated data set, we can sort the firms on the order of the influencing predictors.

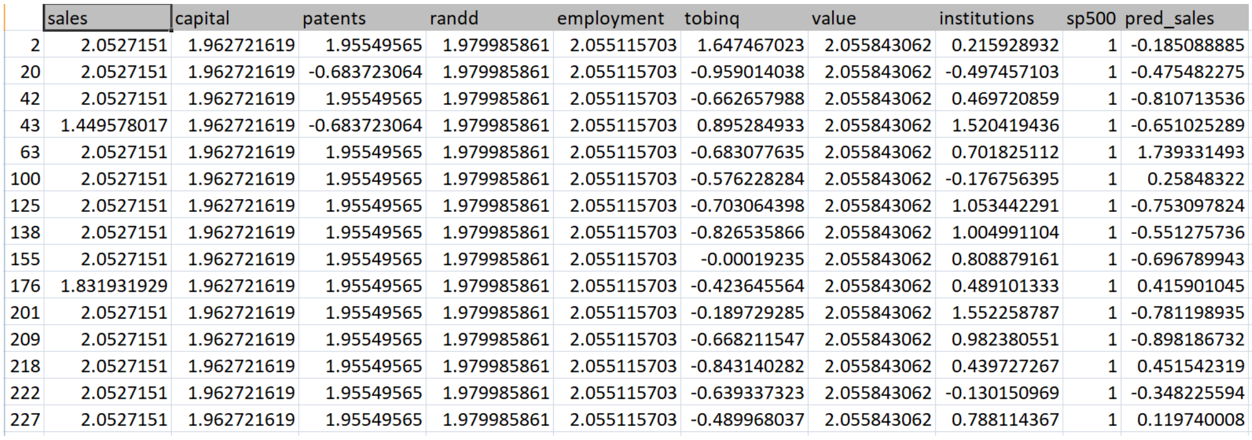
6. From the above steps, we can extract the top 20-50 firms from the data set which might be the safest and most profitable investment options for our company.

## Custom Sort -



| Column | | Sort On | Order |
|---|---|---|---|
| Sort by | employment | Values | Largest to Smallest |
| Then by | value | Values | Largest to Smallest |
| Then by | capital | Values | Largest to Smallest |
| Then by | sp500 | Values | Largest to Smallest |
| Then by | randd | Values | Largest to Smallest |

## Curated Firm Data set -



| | sales | capital | patents | randd | employment | tobinq | value | institutions | sp500 | pred_sales |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | 1.647467023 | 2.055843062 | 0.215928932 | 1 | -0.185088885 |
| 20 | 2.0527151 | 1.962721619 | -0.683723064 | 1.979985861 | 2.055115703 | -0.959014038 | 2.055843062 | -0.497457103 | 1 | -0.475482275 |
| 42 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.662657988 | 2.055843062 | 0.469720859 | 1 | -0.810713536 |
| 43 | 1.449578017 | 1.962721619 | -0.683723064 | 1.979985861 | 2.055115703 | 0.895284933 | 2.055843062 | 1.520419436 | 1 | -0.651025289 |
| 63 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.683077635 | 2.055843062 | 0.701825112 | 1 | 1.739331493 |
| 100 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.576228284 | 2.055843062 | -0.176756395 | 1 | 0.25848322 |
| 125 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.703064398 | 2.055843062 | 1.053442291 | 1 | -0.753097824 |
| 138 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.826535866 | 2.055843062 | 1.004991104 | 1 | -0.551275736 |
| 155 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.00019235 | 2.055843062 | 0.808879161 | 1 | -0.696789943 |
| 176 | 1.831931929 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.423645564 | 2.055843062 | 0.489101333 | 1 | 0.415901045 |
| 201 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.189729285 | 2.055843062 | 1.552258787 | 1 | -0.781198935 |
| 209 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.668211547 | 2.055843062 | 0.982380551 | 1 | -0.898186732 |
| 218 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.843140282 | 2.055843062 | 0.439727267 | 1 | 0.451542319 |
| 222 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.639337323 | 2.055843062 | -0.130150969 | 1 | -0.348225594 |
| 227 | 2.0527151 | 1.962721619 | 1.95549565 | 1.979985861 | 2.055115703 | -0.489968037 | 2.055843062 | 0.788114367 | 1 | 0.119740008 |

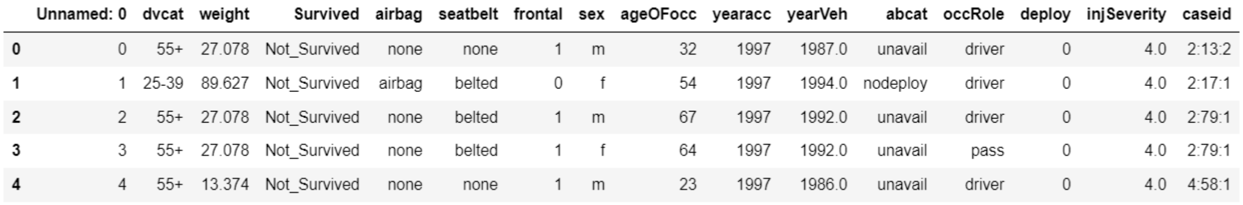
# Problem Statement 2:

# Logistic Regression and Linear Discriminant Analysis

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.

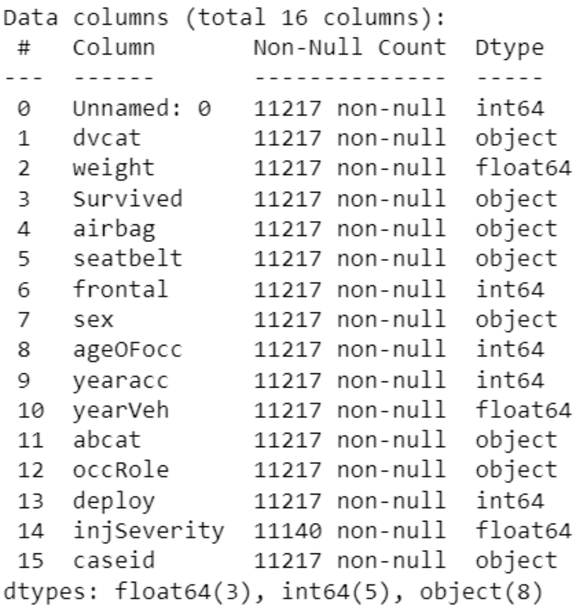
## 2.1) Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis.

### Data Report:



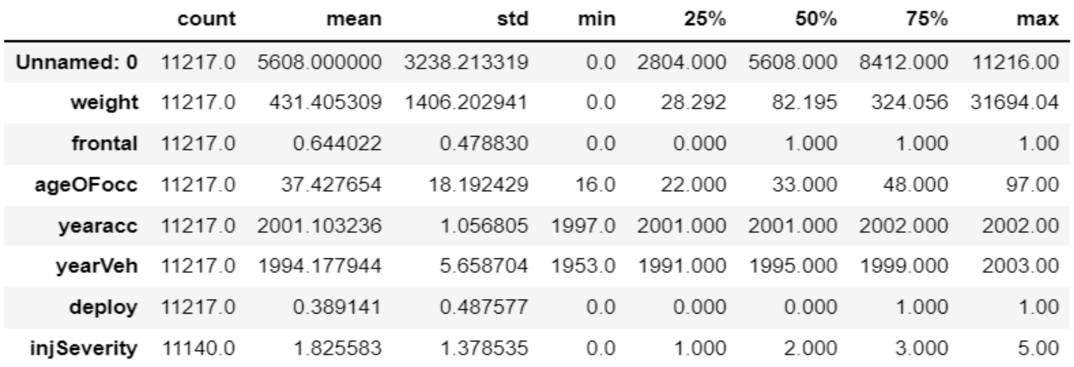
| | Unnamed: 0 | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | yearacc | yearVeh | abcat | occRole | deploy | injSeverity | caseid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 55+ | 27.078 | Not_Survived | none | none | 1 | m | 32 | 1997 | 1987.0 | unavail | driver | 0 | 4.0 | 2:13:2 |
| **1** | 1 | 25-39 | 89.627 | Not_Survived | airbag | belted | 0 | f | 54 | 1997 | 1994.0 | nodeploy | driver | 0 | 4.0 | 2:17:1 |
| **2** | 2 | 55+ | 27.078 | Not_Survived | none | belted | 1 | m | 67 | 1997 | 1992.0 | unavail | driver | 0 | 4.0 | 2:79:1 |
| **3** | 3 | 55+ | 27.078 | Not_Survived | none | belted | 1 | f | 64 | 1997 | 1992.0 | unavail | pass | 0 | 4.0 | 2:79:1 |
| **4** | 4 | 55+ | 13.374 | Not_Survived | none | none | 1 | m | 23 | 1997 | 1986.0 | unavail | driver | 0 | 4.0 | 4:58:1 |

### Data Info:



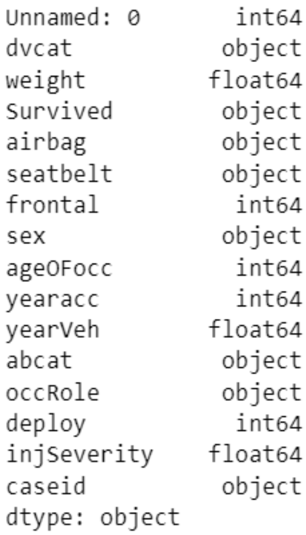
```
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   11217 non-null  int64  
 1   dvcat        11217 non-null  object 
 2   weight       11217 non-null  float64
 3   Survived     11217 non-null  object 
 4   airbag       11217 non-null  object 
 5   seatbelt     11217 non-null  object 
 6   frontal      11217 non-null  int64  
 7   sex          11217 non-null  object 
 8   ageOFocc     11217 non-null  int64  
 9   yearacc      11217 non-null  int64  
 10  yearVeh      11217 non-null  float64
 11  abcat        11217 non-null  object 
 12  occRole      11217 non-null  object 
 13  deploy       11217 non-null  int64  
 14  injSeverity  11140 non-null  float64
 15  caseid       11217 non-null  object 
dtypes: float64(3), int64(5), object(8)
```

## Data description:



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Unnamed: 0** | 11217.0 | 5608.000000 | 3238.213319 | 0.0 | 2804.000 | 5608.000 | 8412.000 | 11216.00 |
| **weight** | 11217.0 | 431.405309 | 1406.202941 | 0.0 | 28.292 | 82.195 | 324.056 | 31694.04 |
| **frontal** | 11217.0 | 0.644022 | 0.478830 | 0.0 | 0.000 | 1.000 | 1.000 | 1.00 |
| **ageOFocc** | 11217.0 | 37.427654 | 18.192429 | 16.0 | 22.000 | 33.000 | 48.000 | 97.00 |
| **yearacc** | 11217.0 | 2001.103236 | 1.056805 | 1997.0 | 2001.000 | 2001.000 | 2002.000 | 2002.00 |
| **yearVeh** | 11217.0 | 1994.177944 | 5.658704 | 1953.0 | 1991.000 | 1995.000 | 1999.000 | 2003.00 |
| **deploy** | 11217.0 | 0.389141 | 0.487577 | 0.0 | 0.000 | 0.000 | 1.000 | 1.00 |
| **injSeverity** | 11140.0 | 1.825583 | 1.378535 | 0.0 | 1.000 | 2.000 | 3.000 | 5.00 |

## Data type check:



```
Unnamed: 0       int64
dvcat           object
weight         float64
Survived        object
airbag          object
seatbelt        object
frontal          int64
sex             object
ageOFocc         int64
yearacc          int64
yearVeh        float64
abcat           object
occRole         object
deploy           int64
injSeverity    float64
caseid          object
dtype: object
```

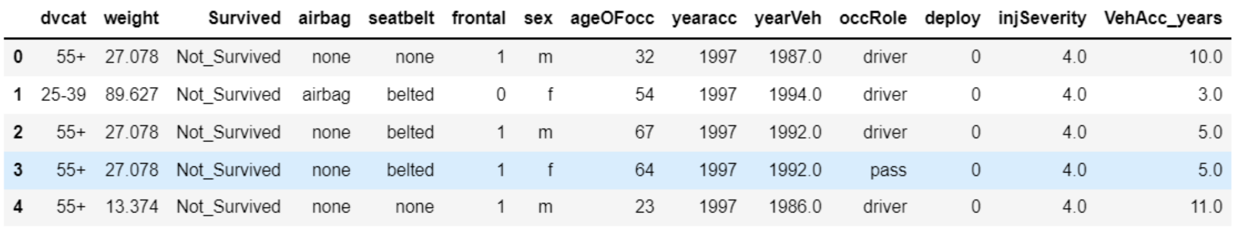
### Insights:

1. We can see that there are 16 fields out of which the target field is the 'Survived'.
2. We observe that there is a field 'Unnamed: 0', which is the unique identifier for the data set. We will have to drop this field.
3. There are 8 object data type fields while the rest of it are int/float.
4. Looking at the description of the data, for the numerical fields, there is a large difference between the values that each field holds, which might require the scaling of the data.
5. There is only one field: 'injSeverity' which contains null values.
6. The data set does not contain any duplicate values.

7. Checking the data type for each field,
8. While checking the data dictionary, we find that the fields 'abcat' and 'deploy' mean the same factor. Hence we can drop one of them.
9. We also see that the field 'caseid' contains unique values about the accident case, hence can be dropped.
10. The data type is sync with the data given in the data set. No field is miscast.

## Feature Engineering:

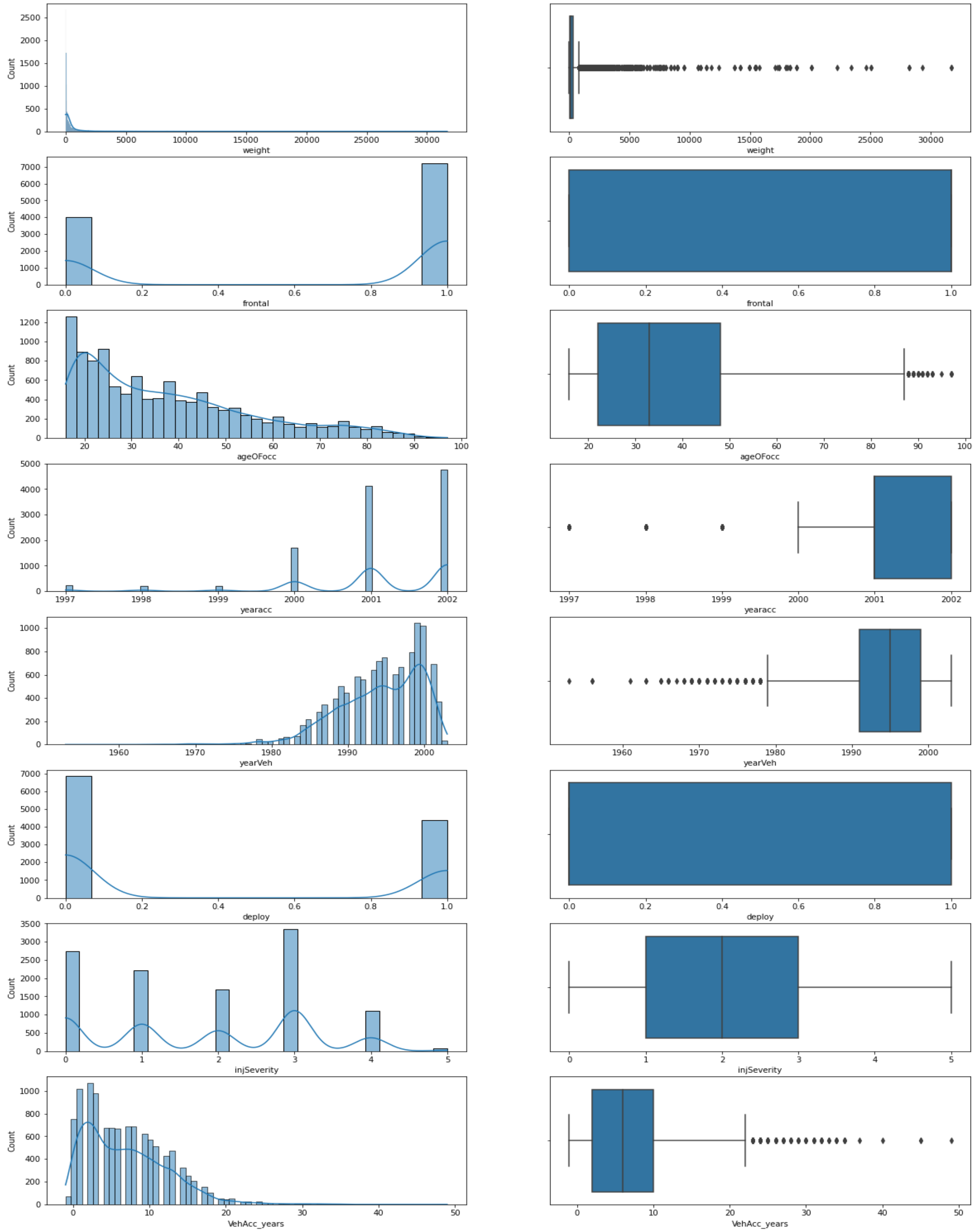
- Removing the unique identifier fields from the data set along with the redundant field, which are ’ Unnamed: 0’,‘caseid’ and ‘abcat’ respectively.
- Creating a field from the difference between 'yearacc' and 'yearVeh', showcasing the number of years before the accident occurred. This will reduce the complexity of modeling with high valued data points (‘yearacc’ and ‘yearVeh’)



| | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | yearacc | yearVeh | occRole | deploy | injSeverity | VehAcc_years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 55+ | 27.078 | Not_Survived | none | none | 1 | m | 32 | 1997 | 1987.0 | driver | 0 | 4.0 | 10.0 |
| 1 | 25-39 | 89.627 | Not_Survived | airbag | belted | 0 | f | 54 | 1997 | 1994.0 | driver | 0 | 4.0 | 3.0 |
| 2 | 55+ | 27.078 | Not_Survived | none | belted | 1 | m | 67 | 1997 | 1992.0 | driver | 0 | 4.0 | 5.0 |
| 3 | 55+ | 27.078 | Not_Survived | none | belted | 1 | f | 64 | 1997 | 1992.0 | pass | 0 | 4.0 | 5.0 |
| 4 | 55+ | 13.374 | Not_Survived | none | none | 1 | m | 23 | 1997 | 1986.0 | driver | 0 | 4.0 | 11.0 |

## Univariate Analysis:

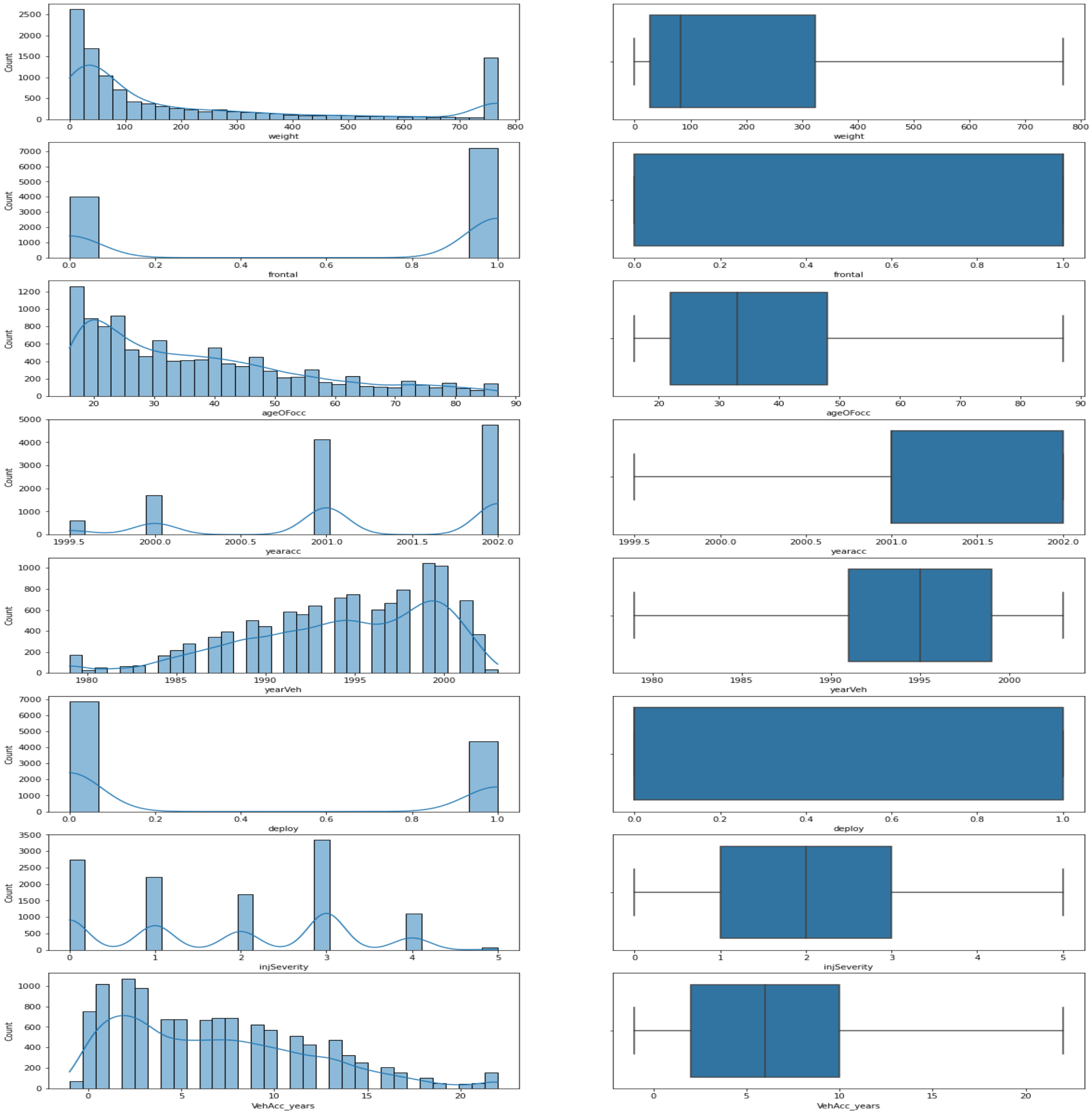
Plotting the count plot and bar plot of all the numerical fields to see the distribution of the field.



**Insights:**

1. We can see that there are a lot of outliers present for the numerical fields resulting in the extreme skewness of the plots. Hence we need to go for outlier treatment.
2. We see that there are about 1400 data points which are posing as outliers for the 'weight' field. Since the number of outliers is about 10% of the total number of data points, we can use capping for treating the outliers.

## After Outlier treatment:



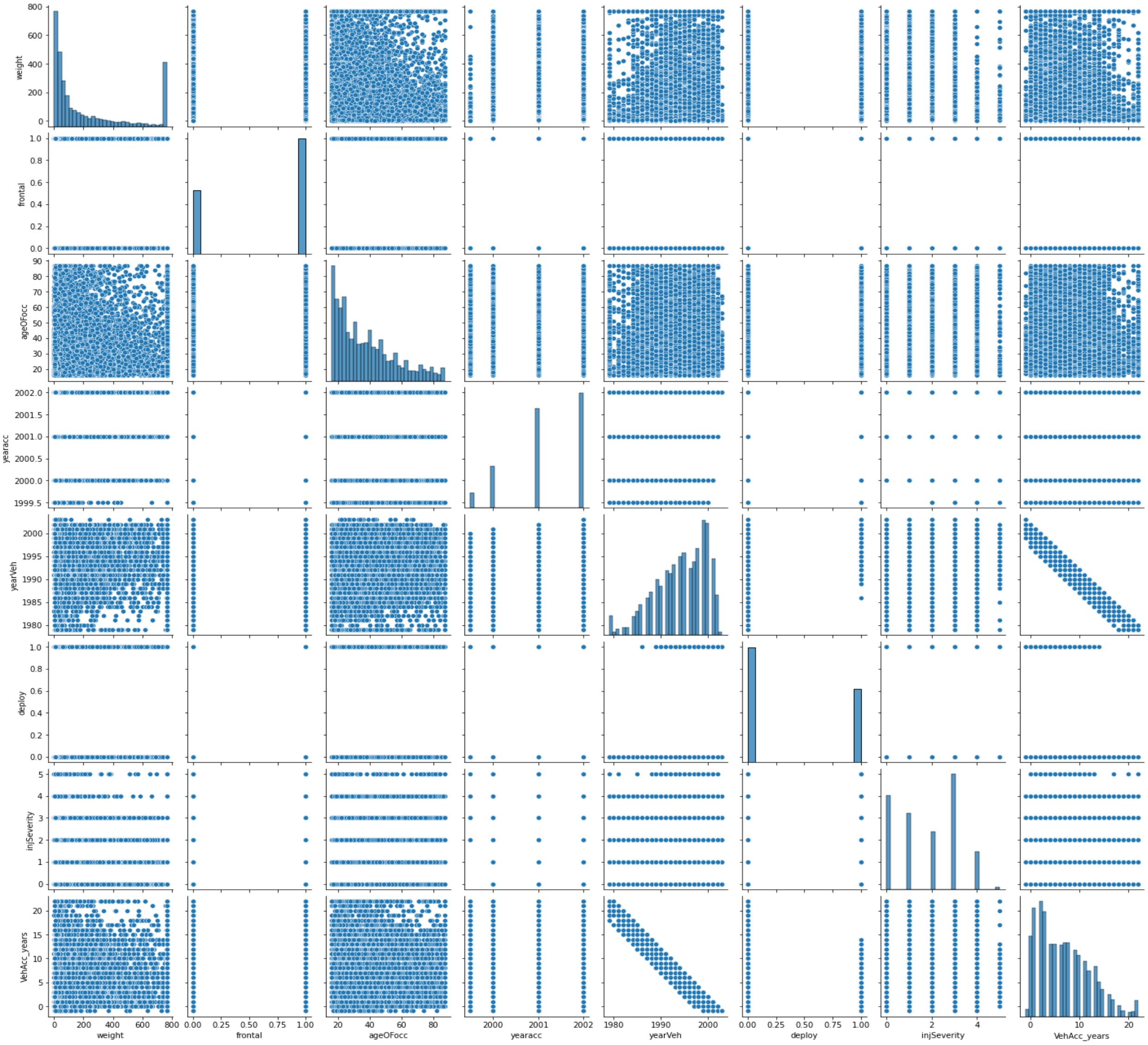
From the new bar plots, we can see that all the outliers have been treated successfully.

**Insights from Univariate Analysis:**

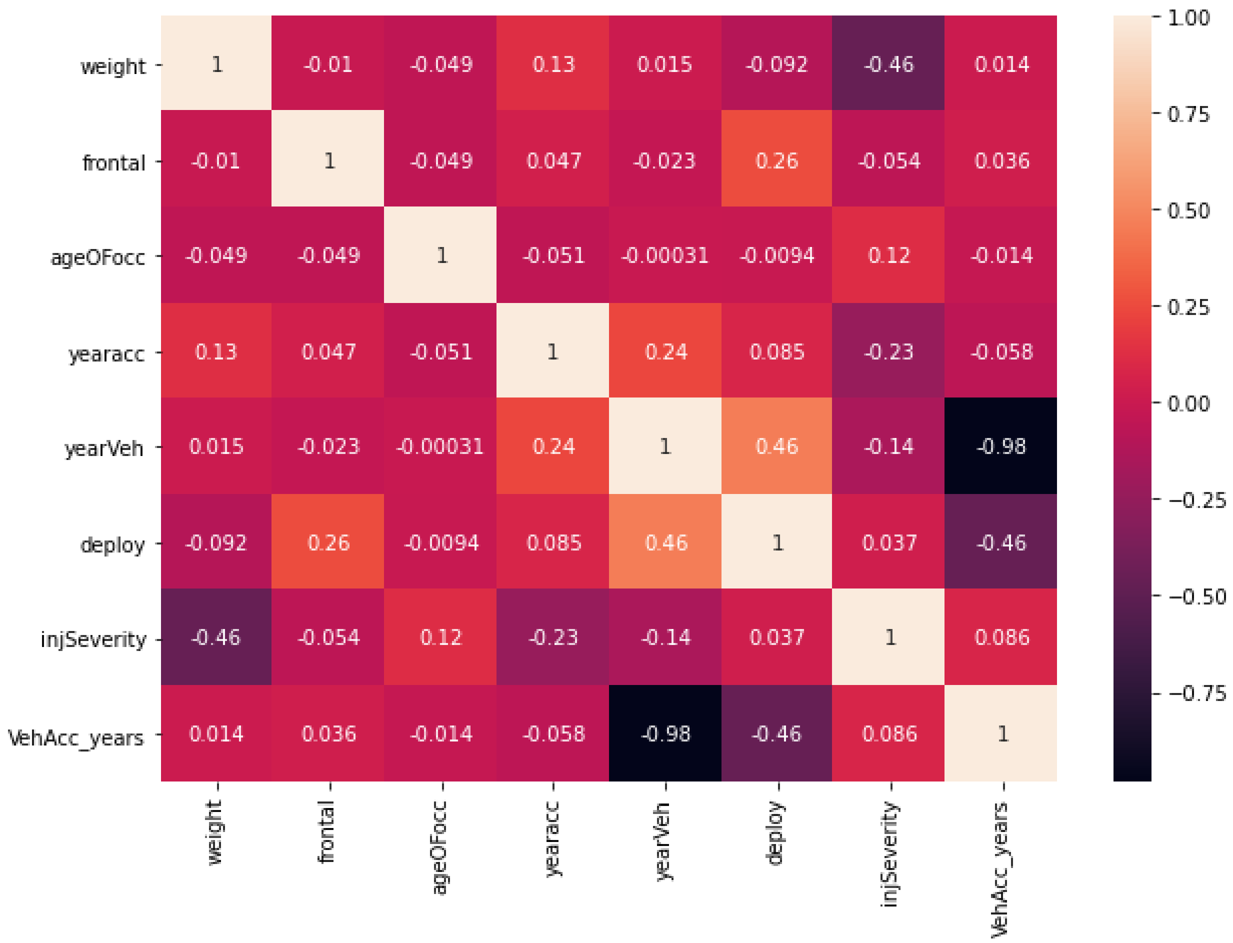
1. From the 'weight' plot, we can see that majority vehicles have weight less than or equal to 300.
2. From the 'ageOFocc' plot, we see that majority of the occupants' lie between 20 to 40.
3. From the 'yearacc' plot, we observe that the accident count has been increasing for every subsequent years.
4. From the 'yearVeh' plot, we can see that highest number of the vehicles have been made around the year 2000.
5. From the 'deploy' plot, the cases where airbags did not deploy are higher than the ones where they deployed, by 60%.
6. The injury severity is highest for the third category (incapacity).
7. From the 'VehAcc_years', the cases are the highest for a difference of 10years or lesser.

# Bivariate Analysis:

## Pairplot-



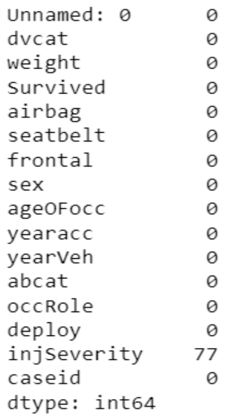
Heatmap-



**Insights:**

1. We see a decent negative correlation betwee injSeverity and weight, hinting that higher the weight of the vehicle, the less the injury severity.
2. We see a good negative correlation between VehAcc_years and deploy, showcasing that as the age of vehicle increases, the cases where the airbags deploy decreases.
3. We see a slight negative correlation between injSeverity and yearacc, telling us that as the years progress, the injury severity has decreased.
4. We see a decent positive correlation between frontal and deploy, telling us that the airbags were deployed more for the cases of frontal impacts.

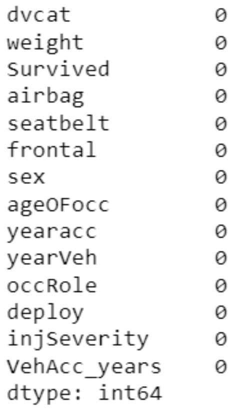
## Treating Null Values:



```
Unnamed: 0      0
dvcat           0
weight          0
Survived        0
airbag          0
seatbelt        0
frontal         0
sex             0
ageOFocc        0
yearacc         0
yearVeh         0
abcat           0
occRole         0
deploy          0
injSeverity    77
caseid          0
dtype: int64
```

We treat the null values by imputing them with the **median** value for the field.

## After Null value treatment:

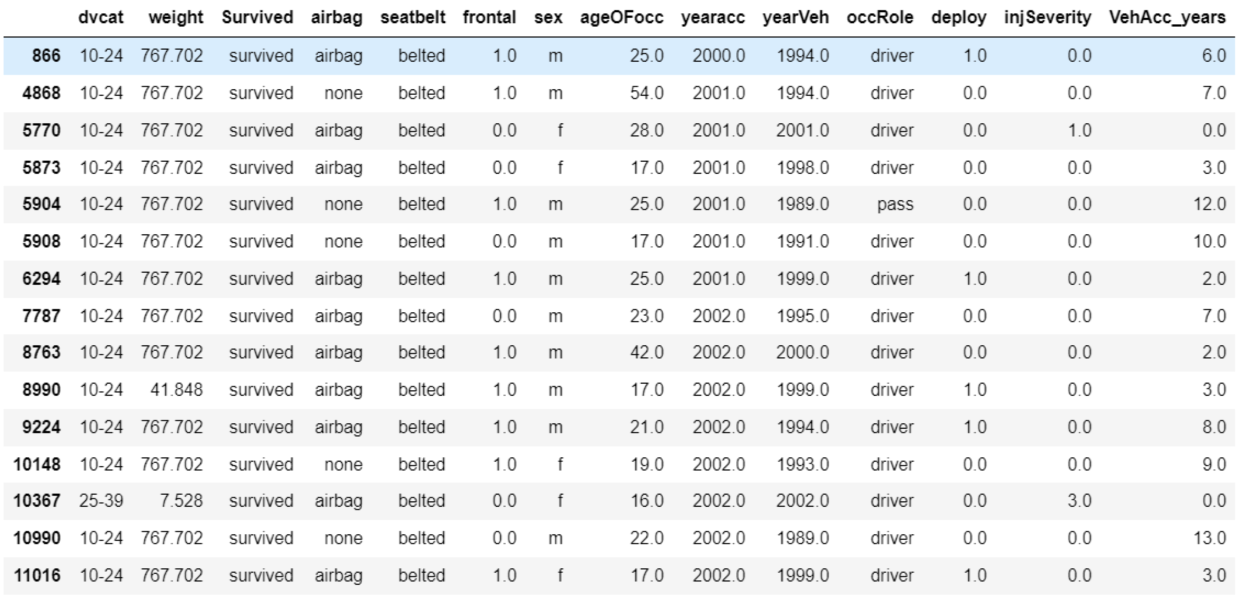


```
dvcat           0
weight          0
Survived        0
airbag          0
seatbelt        0
frontal         0
sex             0
ageOFocc        0
yearacc         0
yearVeh         0
occRole         0
deploy          0
injSeverity     0
VehAcc_years    0
dtype: int64
```

We see that all the null values have been treated successfully.

## Treating Duplicated Values:

When we check for the duplicated values, we see that there are a total of 15 data points which are duplicated-

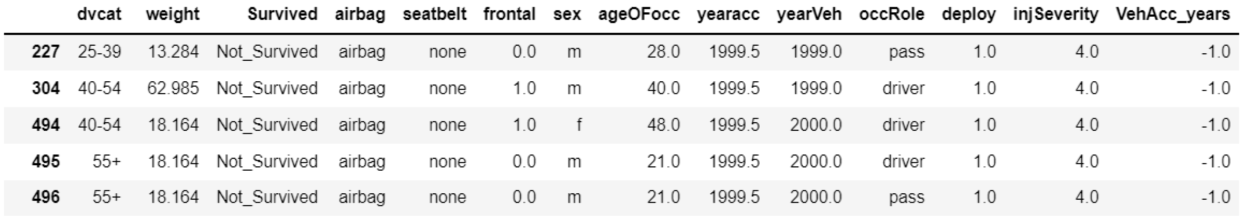


| | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | yearacc | yearVeh | occRole | deploy | injSeverity | VehAcc_years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **866** | 10-24 | 767.702 | survived | airbag | belted | 1.0 | m | 25.0 | 2000.0 | 1994.0 | driver | 1.0 | 0.0 | 6.0 |
| **4868** | 10-24 | 767.702 | survived | none | belted | 1.0 | m | 54.0 | 2001.0 | 1994.0 | driver | 0.0 | 0.0 | 7.0 |
| **5770** | 10-24 | 767.702 | survived | airbag | belted | 0.0 | f | 28.0 | 2001.0 | 2001.0 | driver | 0.0 | 1.0 | 0.0 |
| **5873** | 10-24 | 767.702 | survived | airbag | belted | 0.0 | f | 17.0 | 2001.0 | 1998.0 | driver | 0.0 | 0.0 | 3.0 |
| **5904** | 10-24 | 767.702 | survived | none | belted | 1.0 | m | 25.0 | 2001.0 | 1989.0 | pass | 0.0 | 0.0 | 12.0 |
| **5908** | 10-24 | 767.702 | survived | none | belted | 0.0 | m | 17.0 | 2001.0 | 1991.0 | driver | 0.0 | 0.0 | 10.0 |
| **6294** | 10-24 | 767.702 | survived | airbag | belted | 1.0 | m | 25.0 | 2001.0 | 1999.0 | driver | 1.0 | 0.0 | 2.0 |
| **7787** | 10-24 | 767.702 | survived | airbag | belted | 0.0 | m | 23.0 | 2002.0 | 1995.0 | driver | 0.0 | 0.0 | 7.0 |
| **8763** | 10-24 | 767.702 | survived | airbag | belted | 1.0 | m | 42.0 | 2002.0 | 2000.0 | driver | 0.0 | 0.0 | 2.0 |
| **8990** | 10-24 | 41.848 | survived | airbag | belted | 1.0 | m | 17.0 | 2002.0 | 1999.0 | driver | 1.0 | 0.0 | 3.0 |
| **9224** | 10-24 | 767.702 | survived | airbag | belted | 1.0 | m | 21.0 | 2002.0 | 1994.0 | driver | 1.0 | 0.0 | 8.0 |
| **10148** | 10-24 | 767.702 | survived | none | belted | 1.0 | f | 19.0 | 2002.0 | 1993.0 | driver | 0.0 | 0.0 | 9.0 |
| **10367** | 25-39 | 7.528 | survived | airbag | belted | 0.0 | f | 16.0 | 2002.0 | 2002.0 | driver | 0.0 | 3.0 | 0.0 |
| **10990** | 10-24 | 767.702 | survived | none | belted | 0.0 | m | 22.0 | 2002.0 | 1989.0 | driver | 0.0 | 0.0 | 13.0 |
| **11016** | 10-24 | 767.702 | survived | airbag | belted | 1.0 | f | 17.0 | 2002.0 | 1999.0 | driver | 1.0 | 0.0 | 3.0 |

The duplicated data points have been dropped and handled.

## Treating Anomaly Values:

For the field 'VehAcc_years', we see there is a negative value which is not possible since accidents cannot happen before vehicle manufacture; hence we need to remove the anomaly values.

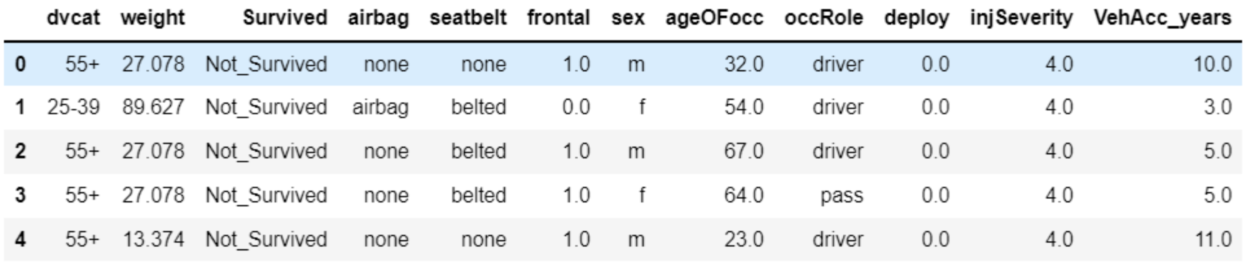


| | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | yearacc | yearVeh | occRole | deploy | injSeverity | VehAcc_years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **227** | 25-39 | 13.284 | Not_Survived | airbag | none | 0.0 | m | 28.0 | 1999.5 | 1999.0 | pass | 1.0 | 4.0 | -1.0 |
| **304** | 40-54 | 62.985 | Not_Survived | airbag | none | 1.0 | m | 40.0 | 1999.5 | 1999.0 | driver | 1.0 | 4.0 | -1.0 |
| **494** | 40-54 | 18.164 | Not_Survived | airbag | none | 1.0 | f | 48.0 | 1999.5 | 2000.0 | driver | 1.0 | 4.0 | -1.0 |
| **495** | 55+ | 18.164 | Not_Survived | airbag | none | 0.0 | m | 21.0 | 1999.5 | 2000.0 | driver | 1.0 | 4.0 | -1.0 |
| **496** | 55+ | 18.164 | Not_Survived | airbag | none | 0.0 | m | 21.0 | 1999.5 | 2000.0 | pass | 1.0 | 4.0 | -1.0 |

We impute the negative value -1 with 0, assuming that the vehicle model year and the accident occurrence year is co-inciding. Hence in this way, the anomaly values have been handled.

(Note: Since we have created a field with the difference between 'yearVeh' and 'yearacc' as 'VehAcc_years', we can drop the original columns)

Our data is ready for the encoding stage:

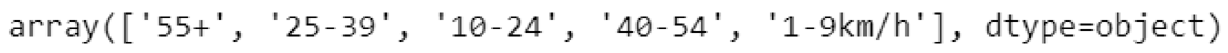


| | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | occRole | deploy | injSeverity | VehAcc_years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 55+ | 27.078 | Not_Survived | none | none | 1.0 | m | 32.0 | driver | 0.0 | 4.0 | 10.0 |
| **1** | 25-39 | 89.627 | Not_Survived | airbag | belted | 0.0 | f | 54.0 | driver | 0.0 | 4.0 | 3.0 |
| **2** | 55+ | 27.078 | Not_Survived | none | belted | 1.0 | m | 67.0 | driver | 0.0 | 4.0 | 5.0 |
| **3** | 55+ | 27.078 | Not_Survived | none | belted | 1.0 | f | 64.0 | pass | 0.0 | 4.0 | 5.0 |
| **4** | 55+ | 13.374 | Not_Survived | none | none | 1.0 | m | 23.0 | driver | 0.0 | 4.0 | 11.0 |

## 2.2) 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).

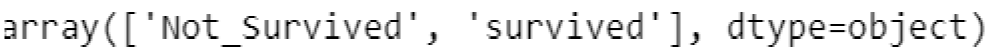
We have a number of categorical fields which are already encoded (‘frontal’,’ deploy’,’ injSeverity’) and some other fields which are yet to be converted.

**'dvcat'-**



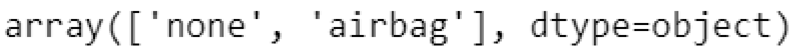
```
array(['55+', '25-39', '10-24', '40-54', '1-9km/h'], dtype=object)
```

**'Survived'-**



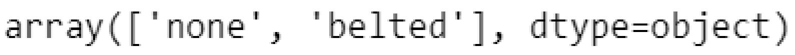
```
array(['Not_Survived', 'survived'], dtype=object)
```

**'airbag'-**



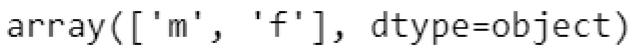
```
array(['none', 'airbag'], dtype=object)
```

**‘seatbelt’-**



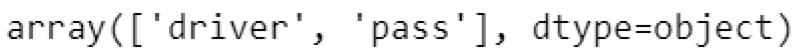
```
array(['none', 'belted'], dtype=object)
```

**‘sex’-**



```
array(['m', 'f'], dtype=object)
```

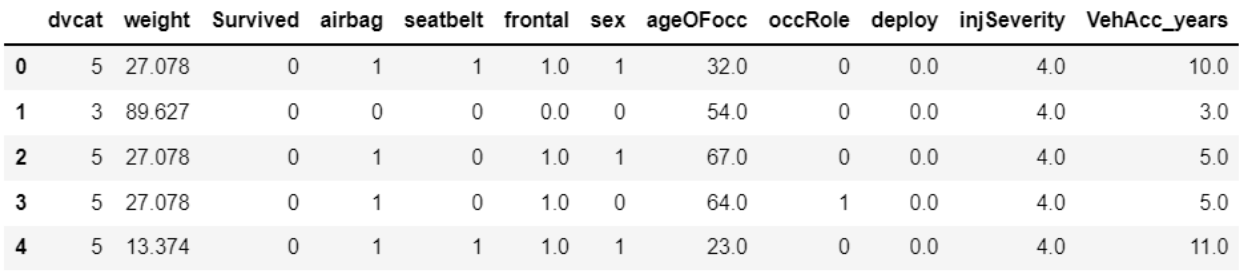
**‘occRole’-**



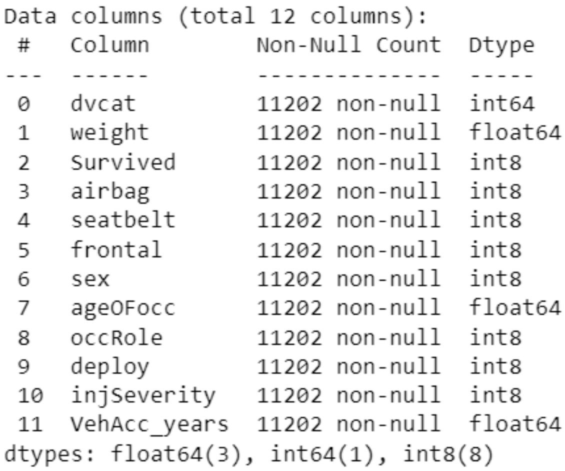
```
array(['driver', 'pass'], dtype=object)
```

'dvcat' is an ordinal categorical value, rest of all the categorical fields are nominal. Hence we use ordinal encoding for ‘dvcat’ and nominal encoding for rest of the categorical fields.

## After Encoding:



| | dvcat | weight | Survived | airbag | seatbelt | frontal | sex | ageOFocc | occRole | deploy | injSeverity | VehAcc_years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 5 | 27.078 | 0 | 1 | 1 | 1.0 | 1 | 32.0 | 0 | 0.0 | 4.0 | 10.0 |
| **1** | 3 | 89.627 | 0 | 0 | 0 | 0.0 | 0 | 54.0 | 0 | 0.0 | 4.0 | 3.0 |
| **2** | 5 | 27.078 | 0 | 1 | 0 | 1.0 | 1 | 67.0 | 0 | 0.0 | 4.0 | 5.0 |
| **3** | 5 | 27.078 | 0 | 1 | 0 | 1.0 | 0 | 64.0 | 1 | 0.0 | 4.0 | 5.0 |
| **4** | 5 | 13.374 | 0 | 1 | 1 | 1.0 | 1 | 23.0 | 0 | 0.0 | 4.0 | 11.0 |



```
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   dvcat         11202 non-null  int64
 1   weight        11202 non-null  float64
 2   Survived      11202 non-null  int8
 3   airbag        11202 non-null  int8
 4   seatbelt      11202 non-null  int8
 5   frontal       11202 non-null  int8
 6   sex           11202 non-null  int8
 7   ageOFocc      11202 non-null  float64
 8   occRole       11202 non-null  int8
 9   deploy        11202 non-null  int8
 10  injSeverity   11202 non-null  int8
 11  VehAcc_years  11202 non-null  float64
dtypes: float64(3), int64(1), int8(8)
```

All the categorical fields have been encoded successfully and our data is now model ready.

## Modelling:

We divide our data set into 2 parts, namely, train data set and test data set. We keep 70% of the data for the train set and 30% for the test set.

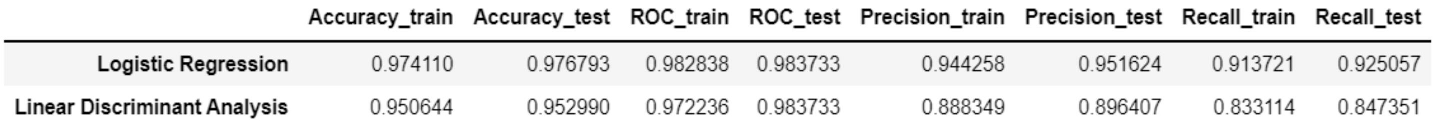
We apply Logistic Regression and Linear Discriminant Analysis(LDA) models on our train data set to train these 2 models.

We have trained our logistic regression and LDA models with the train data and our models are now ready to be tested on the train and test data and their performance metrics are to be measured.

## 2.3) 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.

For the performance metrics, we are using numerous metrics like **Accuracy, ROC_AUC score, Precision, Recall, ROC curve** and **Confusion Matrix** and comparing the values for train and test data set for both the logistic as well as LDA model.

### Result Comparison:



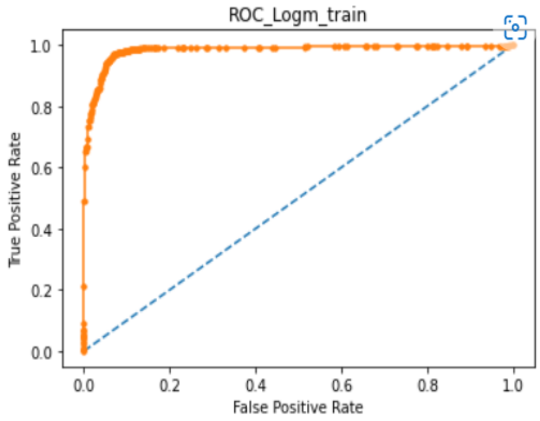
| | Accuracy_train | Accuracy_test | ROC_train | ROC_test | Precision_train | Precision_test | Recall_train | Recall_test |
|---|---|---|---|---|---|---|---|---|
| **Logistic Regression** | 0.974110 | 0.976793 | 0.982838 | 0.983733 | 0.944258 | 0.951624 | 0.913721 | 0.925057 |
| **Linear Discriminant Analysis** | 0.950644 | 0.952990 | 0.972236 | 0.983733 | 0.888349 | 0.896407 | 0.833114 | 0.847351 |

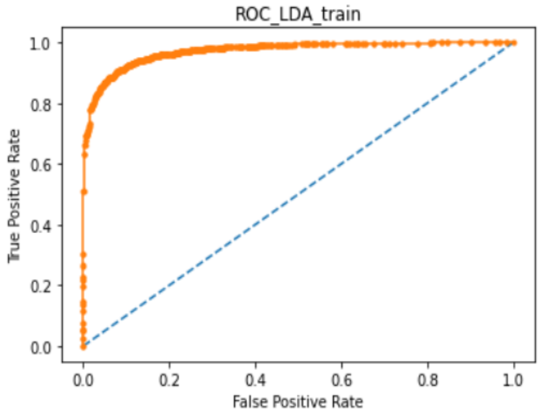
**Insights:**

1. For both the models, the accuracy scores are very close for train and test data, hence implying that both the models are neither under fit nor over fit.
2. When we compare both the classification models, we see that on the basis of the Accuracy score the Logistic regression does better than the LDA model in both the train and the test data set.
3. We see that the Logistic regression model fairs better for test data than the train data.
4. Comparing the ROC scores for train and test, we again see that Logistic regression model does slightly better than LDA for train data set.
5. When it comes to precision and recall, we again see that the logistic model does marginally better than the LDA model for both the train and the test data set.

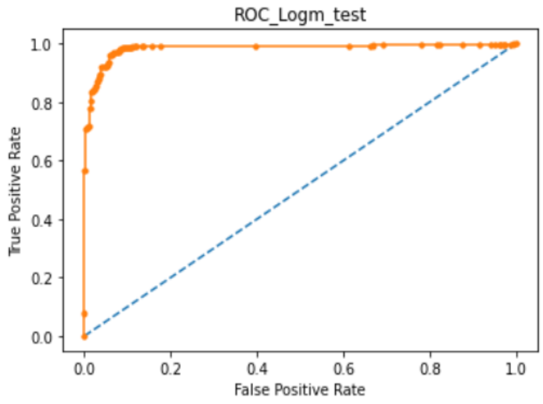
## ROC Curve Comparison:

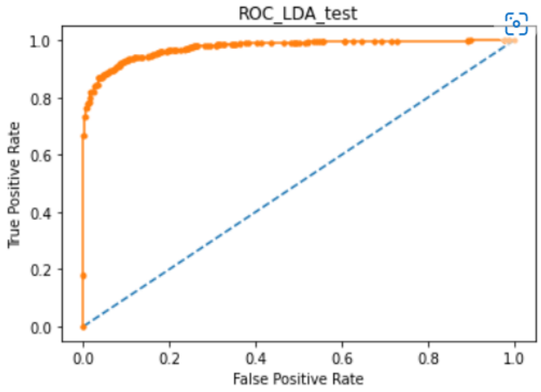
### For train data set-





### For test data set-

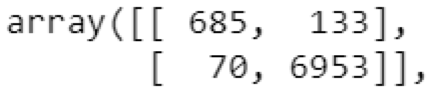




## Confusion Matrix Comparison:

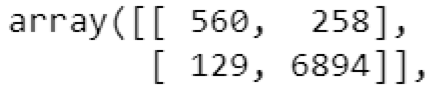
**For train set-**

Logistic Regression-



```
array([[ 685,  133],
       [  70, 6953]],
```

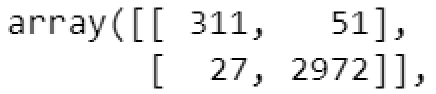
LDA-



```
array([[ 560,  258],
       [ 129, 6894]],
```

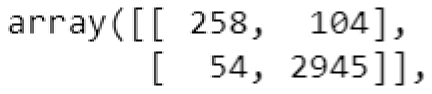
**For test set-**

Logistic Regression-



```
array([[ 311,   51],
       [  27, 2972]],
```

LDA-



```
array([[ 258,  104],
       [  54, 2945]],
```

**Insights:**

1. From the ROC curve, we see that for the train data set, the roc curve for logistic regression is better since the curve is closer to the upper left corner (0,1).
2. Also, for the test data set, the logistic model has a better roc curve.
3. From the confusion matrix, we see that for both the train and test set, the logistic regression model has higher number of true positives and false negatives than the LDA model.

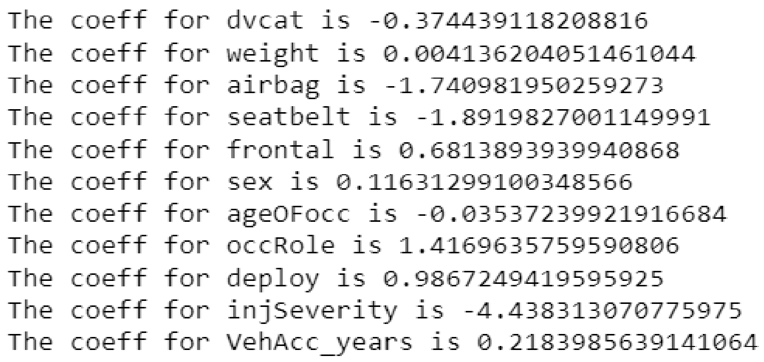
## Final Model Interpretation:

We observe that the logistic regression model has better performance metrics than LDA model with respect to accuracy, ROC_AUC score, precision, recall, ROC curve as well as confusion matrix true positive and false negative values, hence we will use Logistic regression as our final predictive model.

## 2.4) Inference: Based on these predictions, what are the insights and recommendations.

### Business Insights:

1. Checking the feature importances for the Logistic regression model –



```
The coeff for dvcat is -0.374439118208816
The coeff for weight is 0.004136204051461044
The coeff for airbag is -1.740981950259273
The coeff for seatbelt is -1.8919827001149991
The coeff for frontal is 0.6813893939940868
The coeff for sex is 0.11631299100348566
The coeff for ageOFocc is -0.03537239921916684
The coeff for occRole is 1.4169635759590806
The coeff for deploy is 0.9867249419595925
The coeff for injSeverity is -4.438313070775975
The coeff for VehAcc_years is 0.2183985639141064
```

2. We observe that the top 5 attributes which factor in the predictive modelling are-
   a. injSeverity
   b. seatbelt
   c. airbag
   d. occRole
   e. deploy

3. Our designed model has an accuracy of ~97% and a precision of ~95%, which makes the model extremely rugged on performance with test data.
4. Since **'injSeverity'** is the strongest attribute influencing the survival of a person, we can implement laws for the car manufacturers so that they can increase the protection gear on the car to minimize the injury severity. During accidents, the majority injury happens due to sudden impact of the car to the body or the engine breaks from the bonet case and crashes into the boot space. Hence laws can be made to ensure that there is enough padding on the car interior. Also, the car should be tested for impacts to strengthen the car boot space protection.
5. '**seatbelt'** is the second most influencing attribute to the survival of a person. Car manufacturers should implement systems which make the driver aware of the seatbelt situation and gives out an indicator/alarm. In case the driver does not take the seatbelt indicator seriously, the system should not let the car exceed a low speed limit.

6. **Airbags** should be made mandatory for every car since it also factors into the survival of the person. Laws should be instated which restricts usage of cars without airbags installed.
7. From the data, we observe that ~90% of the people who did not survive the car crash were having the ‘**occRole’** of driver. Hence we can understand that the fatalities have a higher chance of happening for the driver’s seat. Car companies should install extra protective gears and padding for the driver’s seat, for eg. steering wheels can have better padding, seatbelts can be made more reliable, etc.
8. **Airbag** systems should be thoroughly crash tested to see if the airbag deployment is successful for the crash or not. Vehicles should pass a minimum set number of airbag deployments before the cars can be sold off to the consumers.