PROJECT ON PREDICTIVE MODELING

(LINEAR REGRESSION, LOGISTICS REGRESSION AND LDA)

M VALLI RAJA SEKAR

PGPDSBA.O.APR22.C

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.

The Main Aim here is to predict the sales using the 759 firms using the Attributes provided for the data and also to Find the 5 attribute that are most Important for prediction of the data.Here Nine Datasets are Provided

Data Dictionary for Firm_level_data:

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.

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. (8 marks)

The shape of the Data contains of total 759 rows and 9 coloumns

The Header of the Data after dropping unnamed will be as follow

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

The Data contains one object as SP500 which is Categorical in Nature,Whereas the rest of the eight data types are either in Integer or Float in Nature.In which we can clearly Identify as there is missing Values present in tubing which need to be Identified and Treated

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sales         759 non-null    float64
 1   capital       759 non-null    float64
 2   patents       759 non-null    int64
 3   randd         759 non-null    float64
 4   employment    759 non-null    float64
 5   sp500         759 non-null    object
 6   tobinq        738 non-null    float64
 7   value         759 non-null    float64
 8   institutions  759 non-null    float64
```

On Identifying the null values, we had observed that the total Null Value Present in data is 21 which can be either treated with Mean or Median

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

Total of 21 missing values in tobinq

Based on the Assumption, The Missing Value is treated with Mean on rest of the Data

```
count    738.000000
mean       2.794910
std        3.366591
min        0.119001
25%        1.018783
50%        1.680303
75%        3.139309
max       20.000000
Name: tobinq, dtype: float64
```

```
count    759.000000
mean       2.794910
std        3.319629
min        0.119001
25%        1.036000
50%        1.741800
75%        3.082979
max       20.000000
Name: tobinq, dtype: float64
```

Before and Treating the Missing Value with Mean, there is not Much changes in the Data. The Both had Attached above

After Checking the Missing Value, the Next Procedure is to check whether Anomalies or Bad Data is present in the data

In Terms of Bad Data, there is no Bad data is present in the Data sent

For Anomalies,

	sales	capital	patents	randd	employment	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000
mean	2689.705158	1977.747498	25.831357	439.938074	14.164519	2.794910	2732.734750	43.020540
std	8722.060124	6466.704896	97.259577	2007.397588	43.321443	3.319629	7071.072362	21.685586
min	0.138000	0.057000	0.000000	0.000000	0.006000	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	4.628262	0.927500	1.036000	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	1.741800	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	3.082979	2054.160386	60.510000
max	135696.788200	93625.200560	1220.000000	30425.255860	710.799925	20.000000	95191.591160	90.150000

The Randd in above data contains lots of 0 present in the Data. Hence treating those with the value of Mean/ Median.We Might get better Predictions of the data instead of dropping those rows

```
count       759.000000
mean        439.938074
std        2007.397588
min           0.000000
25%           4.628262
50%          36.864136
75%         143.253403
max       30425.255860
Name: randd, dtype: float64
```

```
count       759.000000
mean        444.406454
std        2006.452894
min           0.013687
25%          14.634818
50%          36.864136
75%         143.253403
max       30425.255860
Name: randd, dtype: float64
```

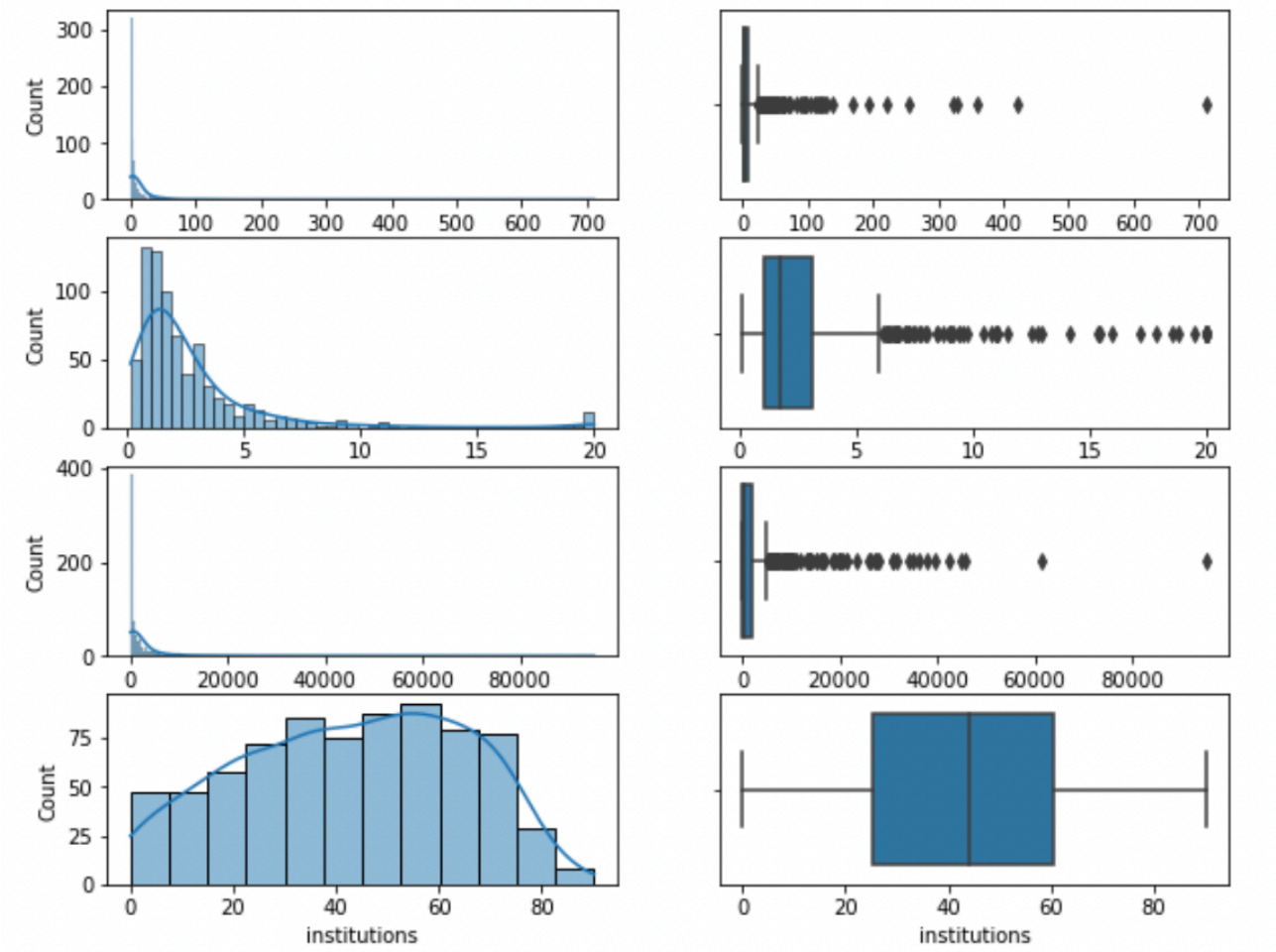
After Making above changes, we can see that only Means has been slightly changed Whereas the the Rest of the Data’s are present

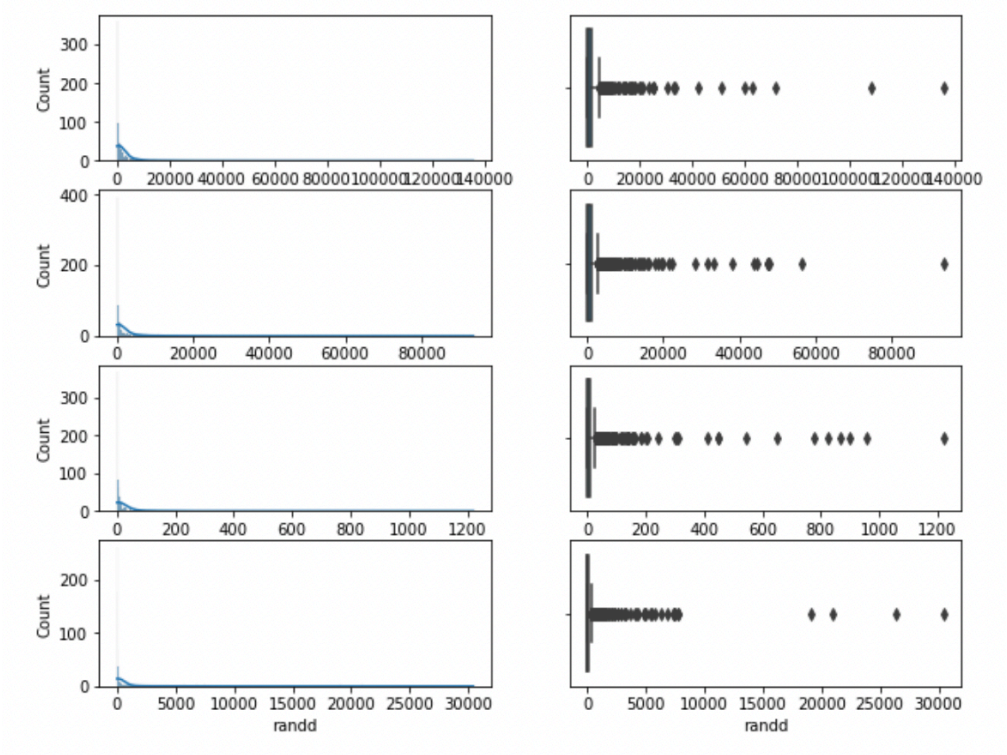
Data Visualization

Univariate Analysis

Univariate data requires to analyze each variable separately. Uni means one, so in other words the data has only one variable.In Univariate Analysis, we can use Histogram and Box plot for Numerical type where as countplot for Categorical type

The Histogram and Box Plot Analysis of the eight numeric data are given below

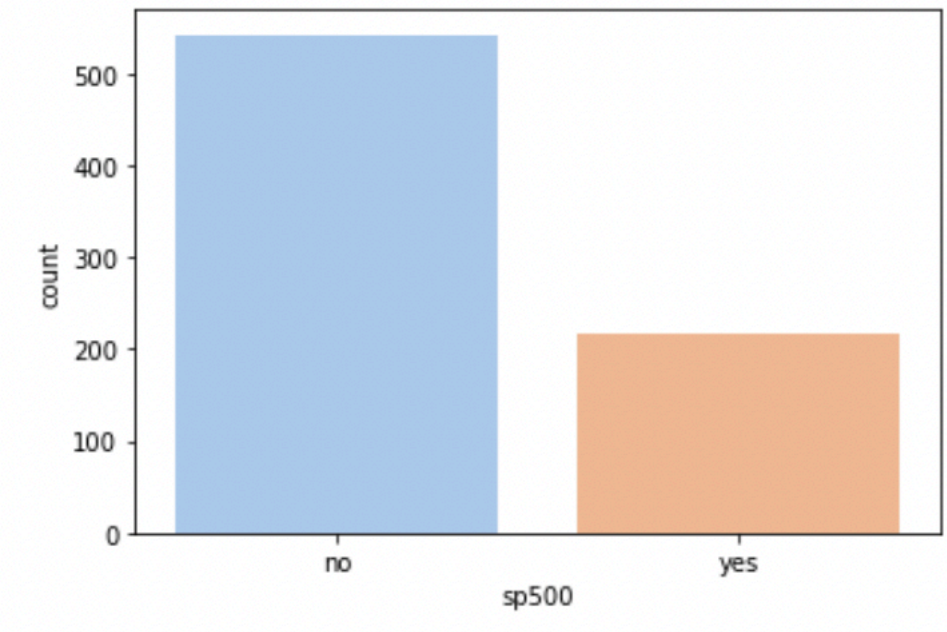




The Eight Numerical Data is presented in the form of Histogram and Boxplot and from the above observation we can clearly see that there is lots of Outlier present in the Data and also Histogram are not much Normally Distributed. Hence Treatment of Outliers and Standardisation is required for the above data.

The Analysis of one Categorical variable of SP500 are as follow

```
no      542
yes     217
Name: sp500, dtype: int64
```



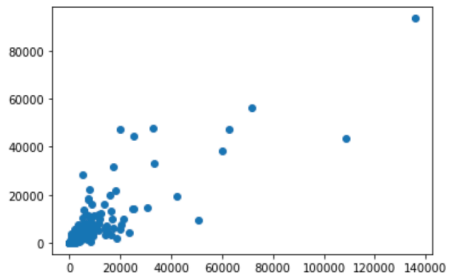
In above data, Only 217 company are part of the SP500 whereas the rest are not part of SP500

Bivariate Analysis

In Bivariate Analysis, there are two variables wherein the analysis is related to cause and the relationship between the two variables.Here the Numeric and Numeric relation are established in the relationship using Pairplot/scatter plot whereas Categorical and Categorical are established with Countplot with Hue

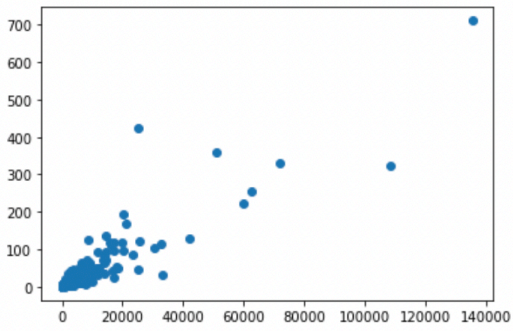
```
plt.scatter(df['sales'],df['capital'])
```

```
<matplotlib.collections.PathCollection at 0x7fd370cd4760>
```



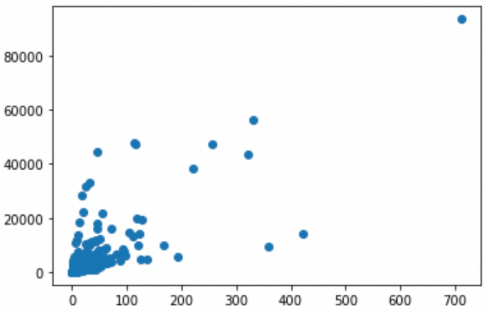

```
plt.scatter(df['sales'],df['employment'])
```

```
<matplotlib.collections.PathCollection at 0x7fd370d
```



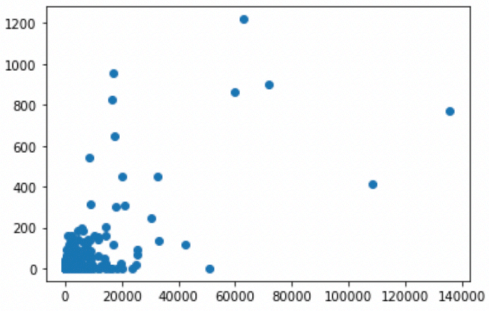

```
plt.scatter(df['employment'],df['capital'])
```

```
<matplotlib.collections.PathCollection at 0x7fd370e7b37
```



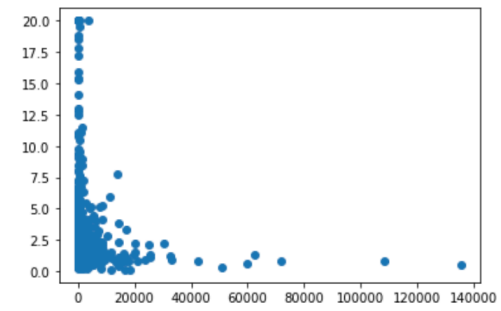

```
plt.scatter(df['sales'],df['patents'])
```

```
<matplotlib.collections.PathCollection at 0x7fd370e
```



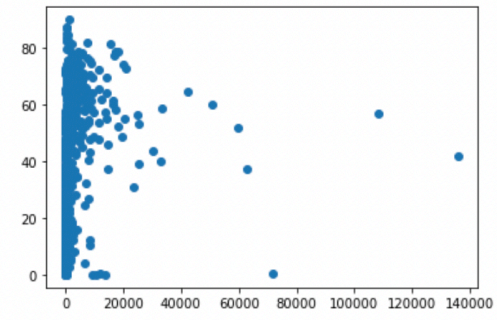

```
plt.scatter(df['sales'],df['tobinq'])
```

```
<matplotlib.collections.PathCollection at 0x7fd3922033
```

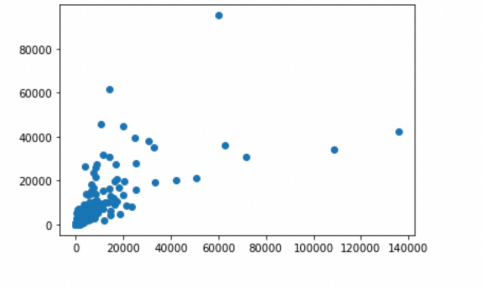


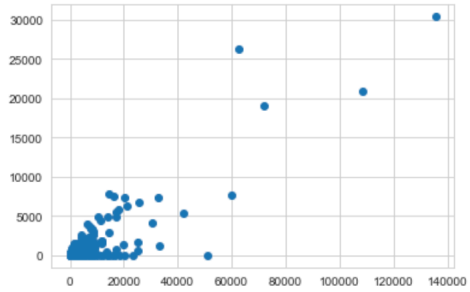

```
plt.scatter(df['sales'],df['institutions'])
```

```
<matplotlib.collections.PathCollection at 0x7fd394cc1a
```



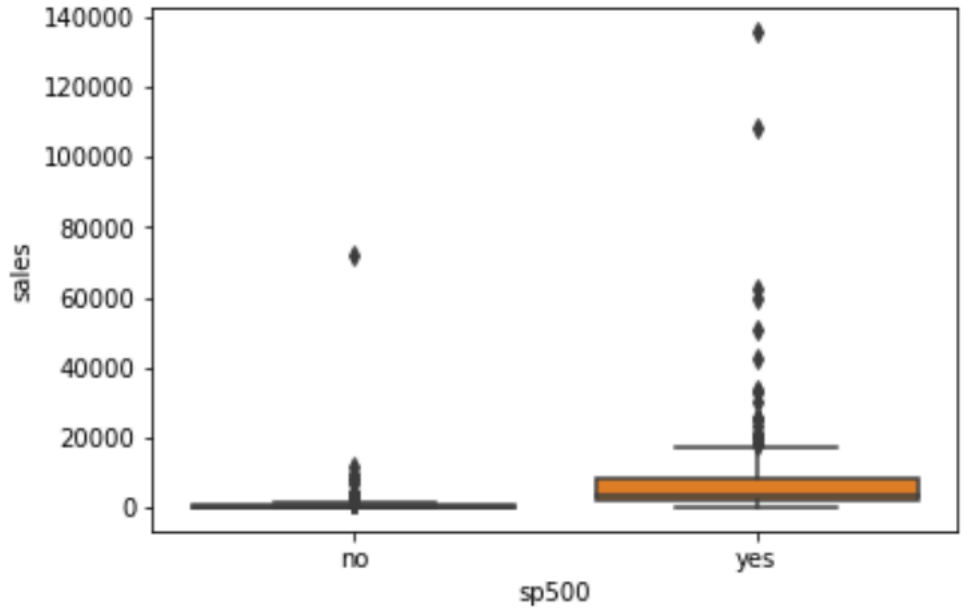
```
plt.scatter(df['sales'],df['value'])
<matplotlib.collections.PathCollection at 0x7fd394cf6
```





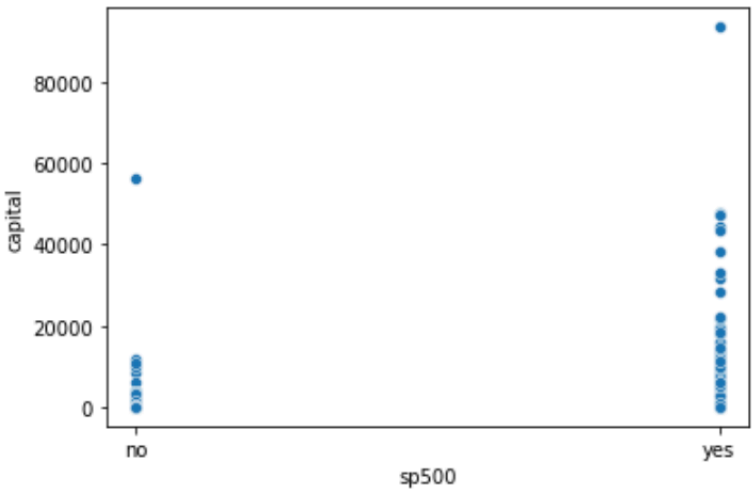
On above analysing the the numerical parameter of the data, we can clearly identify the data Dependent Variable of sales have some kind of correlation ship with all other Independent Variables present in the data ,while the relationship can be better identified while finding Correlation ship and also while using the Heat Map for the data.

```
<AxesSubplot:xlabel='sp500', ylabel='sales'>
```



On above Analysing only one Independent Categorical Variable SP500 with only one Dependent Variable of sales.We can Identify that the sales is higher in SP500 member company than comparing with the the non member of SP500

The other most fascinating thing is that SP500 member company has higher capital compared to Non SP500 companies

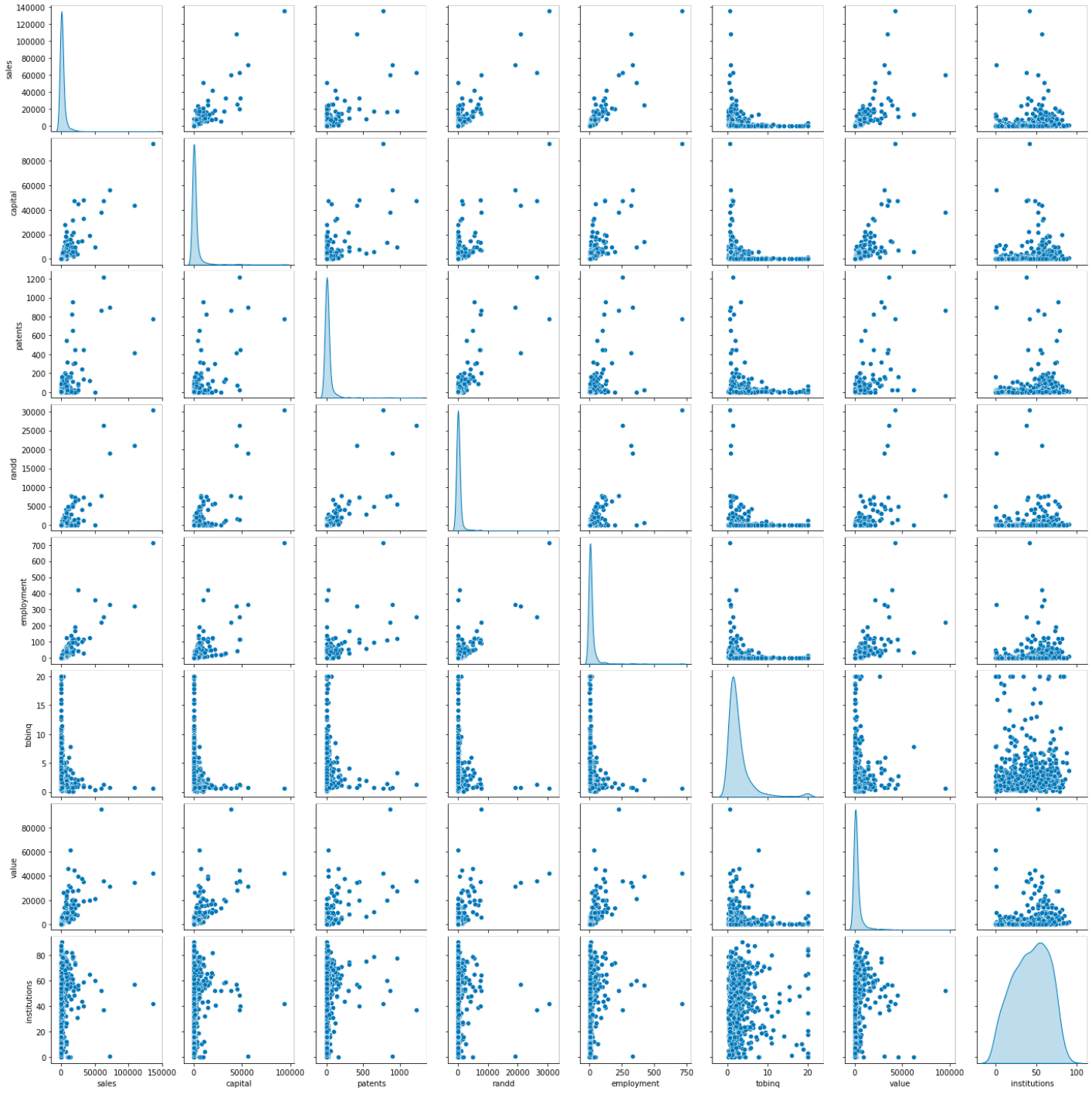


Multivariate Analysis

Multivariate descriptive displays or plots are designed to reveal the relationship among several variables simulataneously.. As was the case when examining relationships among pairs of variables, there are several basic characteristics of the relationship among sets of variables that are of interest. Some of the Interesting Multivariate Analysis are pair plot,Heat Map, Facet etc

Pairwise Analysis

A pairs plot allows us to see both distribution of single variables and relationships between two variable.The Pairwise plots are Mentioned below which determines the Overall relationship between the Dependent Variable and all other Independent variable. From this we Can identify microlevel and we can establish some kind of Relation


sales
capital
patents
randd
employment
tobinq
value
institutions

Heat map

The heat map is a data visualization technique that shows magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity, giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space.

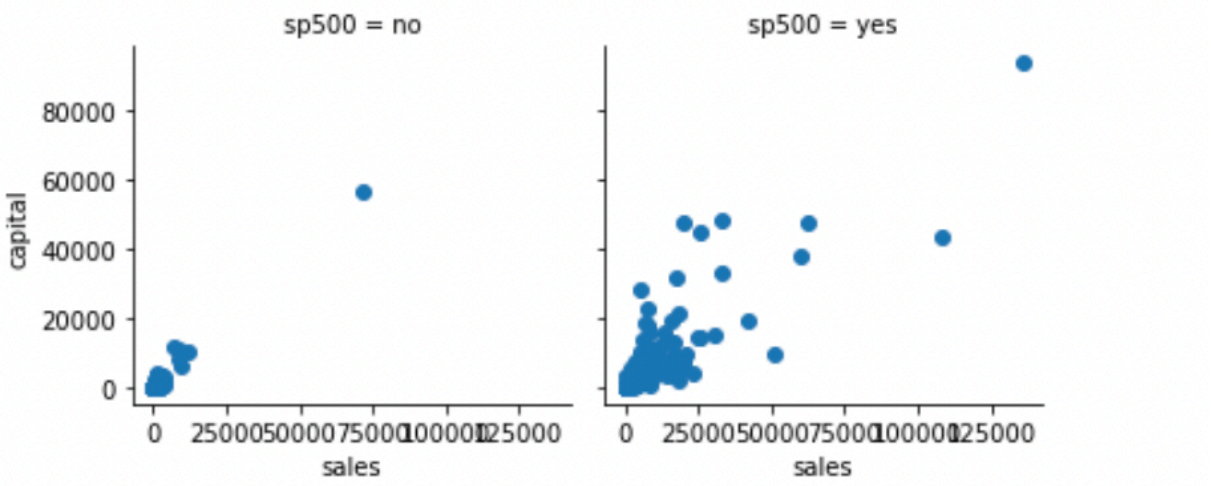
The Data of correlation and Pictorial representation of Heat map are given below

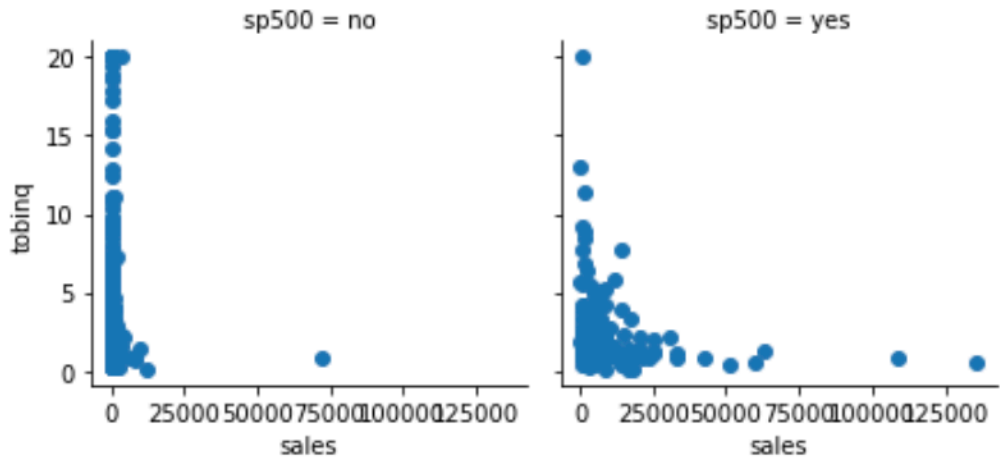
	sales	capital	patents	randd	employment	tobinq	value	institutions
sales	1.000000	0.869595	0.682134	0.870660	0.908868	-0.113349	0.713778	0.116483
capital	0.869595	1.000000	0.633339	0.771046	0.771263	-0.130422	0.715893	0.099160
patents	0.682134	0.633339	1.000000	0.820332	0.626341	-0.049007	0.619547	0.127751
randd	0.870660	0.771046	0.820332	1.000000	0.778555	-0.059051	0.585316	0.053122
employment	0.908868	0.771263	0.626341	0.778555	1.000000	-0.113638	0.668336	0.144300
tobinq	-0.113349	-0.130422	-0.049007	-0.059051	-0.113638	1.000000	-0.006127	-0.024943
value	0.713778	0.715893	0.619547	0.585316	0.668336	-0.006127	1.000000	0.138269
institutions	0.116483	0.099160	0.127751	0.053122	0.144300	-0.024943	0.138269	1.000000

From the Heatmap we can identify the Correlationship of data in better way than establishing the relationship in Univariate and Bivariate Method

Facetgrid

It is most useful when you have two discrete variables, and all combinations of the variables exist in the data.In below data, the higher is the sales, capital and tubing in SP500 compared to non SP500





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

Impute Null Values-

The Data contains one object as SP500 which is Categorical in Nature,Whereas the rest of the eight data types are either in Integer or Float in Nature.In which we can clearly Identify as there is missing Values present in sp500 which need to be Identified and Treated

#	Column	Non-Null Count	Dtype
0	sales	759 non-null	float64
1	capital	759 non-null	float64
2	patents	759 non-null	int64
3	randd	759 non-null	float64
4	employment	759 non-null	float64
5	sp500	759 non-null	object
6	tobinq	738 non-null	float64
7	value	759 non-null	float64
8	institutions	759 non-null	float64

On Identifying the null values, we had observed that the total Null Value Present in data is 21 which can be either treated with Mean or Median

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

Total of 21 missing values in tobinq

Based on the Assumption, The Missing Value of tubinq is with Mean of the Data

Before and Treating the Missing Value with Mean, there is not Much changes in the Data. The Both had Attached above

```
count    738.000000
mean       2.794910
std        3.366591
min        0.119001
25%        1.018783
50%        1.680303
75%        3.139309
max       20.000000
Name: tobinq, dtype: float64
```

```
count    759.000000
mean       2.794910
std        3.319629
min        0.119001
25%        1.036000
50%        1.741800
75%        3.082979
max       20.000000
Name: tobinq, dtype: float64
```

After Checking the Missing Value, the Next Procedure is to check whether Anomalies or Bad Data is present in the data

In Terms of Bad Data, there is no Bad data is present in the Data

For Anomalies,

	sales	capital	patents	randd	employment	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000
mean	2689.705158	1977.747498	25.831357	439.938074	14.164519	2.794910	2732.734750	43.020540
std	8722.060124	6466.704896	97.259577	2007.397588	43.321443	3.319629	7071.072362	21.685586
min	0.138000	0.057000	0.000000	0.000000	0.006000	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	4.628262	0.927500	1.036000	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	1.741800	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	3.082979	2054.160386	60.510000
max	135696.788200	93625.200560	1220.000000	30425.255860	710.799925	20.000000	95191.591160	90.150000

The Randd in above data contains lots of 0 present in the Data. Hence treating those with the value of Mean/ Median.We Might get better Predictions of the data instead of dropping those rows

```
count       759.000000
mean        439.938074
std        2007.397588
min           0.000000
25%           4.628262
50%          36.864136
75%         143.253403
max       30425.255860
Name: randd, dtype: float64
```

```
count       759.000000
mean        444.406454
std        2006.452894
min           0.013687
25%          14.634818
50%          36.864136
75%         143.253403
max       30425.255860
Name: randd, dtype: float64
```

After Making above changes, we can see that only Means has been slightly changed Whereas the the Rest of the Data's are present are almost same

Hence We had Imputed the Null Values inform of Treating With Missing Values, Anamolies and Bad data with either treating Mean of variable

Do you think scaling is necessary in this case

	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000
mean	2689.705158	1977.747498	25.831357	444.406454	14.164519	0.285903	2.794910	2732.734750	43.020540
std	8722.060124	6466.704896	97.259577	2006.452894	43.321443	0.452141	3.319629	7071.072362	21.685586
min	0.138000	0.057000	0.000000	0.013687	0.006000	0.000000	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	14.634818	0.927500	0.000000	1.036000	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	0.000000	1.741800	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	1.000000	3.082979	2054.160386	60.510000
max	135696.788200	93625.200560	1220.000000	30425.255860	710.799925	1.000000	20.000000	95191.591160	90.150000

In above we can clearly see that there is difference in the Mean between their Attributes and also each attributes are measured in the different Parameters. Hence **Scaling is required in the dataset**

On applying the Z score for the data, the Data look

	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000	759.000000
mean	2689.705158	1977.747498	25.831357	444.406454	14.164519	0.285903	2.794910	2732.734750	43.020540
std	8722.060124	6466.704896	97.259577	2006.452894	43.321443	0.452141	3.319629	7071.072362	21.685586
min	0.138000	0.057000	0.000000	0.013687	0.006000	0.000000	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	14.634818	0.927500	0.000000	1.036000	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	0.000000	1.741800	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	1.000000	3.082979	2054.160386	60.510000
max	135696.788200	93625.200560	1220.000000	30425.255860	710.799925	1.000000	20.000000	95191.591160	90.150000

By doing Scaling , The Data got normalised and made mean as 0 and standard Deviation as 1. So by scaling all the data are in Normalised Manner

1.3)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

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   sales          759 non-null    float64
 1   capital        759 non-null    float64
 2   patents        759 non-null    int64
 3   randd          759 non-null    float64
 4   employment     759 non-null    float64
 5   sp500          759 non-null    object
 6   tobinq         738 non-null    float64
 7   value          759 non-null    float64
 8   institutions   759 non-null    float64
```

Now we can see that only SP 500 is the only category value present in the Data. The SP 500 should be converted into either float or Integer by the process of Label Encoding

Now info of the Data are as followed

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 759 entries, 0 to 758
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    int64
 3   randd          759 non-null    float64
 4   employment     759 non-null    float64
 5   sp500          759 non-null    int64
 6   tobinq         759 non-null    float64
 7   value          759 non-null    float64
 8   institutions   759 non-null    float64
dtypes: float64(7), int64(2)
memory usage: 53.5 KB
```

Hence now the whole data contains either float or Integer. Now we can Proceed the Data for Modelling Process

Now we need to split the data into test and Train in 30:70. Which Mean that the data contains 30 % in test data and the rest 70% in Training data.Before Splitting the data into Test and Train, the first test is segregate Dependent Variable Sales from the Rest of the Independent Variable. The Constant should be added to the Variable

Splitting of Data

```
# independent variables
X = df_1.drop(["sales"], axis=1)
# dependent variable
y = df_1[["sales"]]
```

```
X = sm.add_constant(X)
```

Splitting of Data are as follow

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=5
)
```

As earlier the Test and Train data has been splited in the ratio of (30:70) and in the Random state of 5

Applying the Linear regression in the Model using the OLS Method

```
olsmod = sm.OLS(y_train, X_train)
olsres = olsmod.fit()
```

On the Applying the linear regression using the OLS Method

The Model Summary looks like Below

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.928
Model:                            OLS   Adj. R-squared:                  0.927
Method:                 Least Squares   F-statistic:                     837.4
Date:                Thu, 13 Oct 2022   Prob (F-statistic):          3.93e-292
Time:                        19:49:05   Log-Likelihood:                871.52
No. Observations:                 531   AIC:                            -1725.
Df Residuals:                     522   BIC:                            -1687.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0096      0.008      1.181      0.238      -0.006       0.026
capital        0.2816      0.034      8.357      0.000       0.215       0.348
patents       -0.0162      0.032     -0.512      0.609      -0.078       0.046
randd          0.1127      0.060      1.888      0.060      -0.005       0.230
employment     0.4010      0.024     16.469      0.000       0.353       0.449
sp500          0.0038      0.003      1.130      0.259      -0.003       0.010
tobinq        -0.0161      0.005     -3.323      0.001      -0.026      -0.007
value          0.2080      0.020     10.149      0.000       0.168       0.248
institutions   0.0011      0.002      0.476      0.634      -0.004       0.006
==============================================================================
Omnibus:                      217.262   Durbin-Watson:                   1.908
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1550.939
Skew:                           1.625   Prob(JB):                         0.00
Kurtosis:                      10.716   Cond. No.                         37.4
==============================================================================
```

In the Above Model R- Squared and Adj R-Squared Value is 0.928 and 0.927. But lots of attributes Present in the data has P-value >=0.5 hence most of Attributes become less Significant in this case. So the Multicollinearity using the VIF should be checked and select the last 5 best attributes

```
VIF values:

const           15.770770
capital          5.911542
patents          2.589940
randd            3.109807
employment       4.975397
sp500            2.737928
tobinq           1.502707
value            6.076885
institutions     1.254328
dtype: float64
```

In case of VIF for Multicollinearity greater than 4 or 5 can make the curse of Dimensionality. So the Value of VIF should be treated and need to be dropped and then nessacary actions can be taken

The Below Scenario Can be explain the for best scenario of the data Modelling

Scenario 1

Here in scenario 1 our focus will be getting the 5 best attributes based on expert Opinion of the Data and deducting the nessacary based on our requirement with based on Expert

At first by dropping the Value, the data summary and VIF looks alike

```
olsmod_1 = sm.OLS(y_train, X_train_1)
olsres_1 = olsmod_1.fit()
print(olsres_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.913
Model:                            OLS   Adj. R-squared:                  0.912
Method:                 Least Squares   F-statistic:                     788.6
Date:                Mon, 17 Oct 2022   Prob (F-statistic):          3.33e-273
Time:                        12:38:01   Log-Likelihood:                823.71
No. Observations:                 531   AIC:                            -1631.
Df Residuals:                     523   BIC:                            -1597.
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            0.0305      0.009      3.539      0.000       0.014       0.047
capital          0.4866      0.029     16.495      0.000       0.429       0.545
patents         -0.0348      0.035     -1.008      0.314      -0.103       0.033
randd            0.1762      0.065      2.715      0.007       0.049       0.304
employment       0.4665      0.026     18.178      0.000       0.416       0.517
sp500            0.0105      0.004      2.933      0.004       0.003       0.018
tobinq           0.0071      0.005      1.510      0.132      -0.002       0.016
institutions     0.0018      0.003      0.715      0.475      -0.003       0.007
==============================================================================
Omnibus:                      237.127   Durbin-Watson:                   1.893
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2163.054
Skew:                           1.718   Prob(JB):                         0.00
Kurtosis:                      12.271   Cond. No.                         36.9
==============================================================================
```

```
VIF values:

const           14.765009
capital          3.789617
patents          2.581241
randd            3.075643
employment       4.625513
sp500            2.630212
tobinq           1.169457
institutions     1.253192
dtype: float64
```

Here after dropping the values, the R squared value dropped to 91.3% and the VIF values is less than 5 but we can have some p-values greater than 0.5

So we can now retain institutions and drop the tobinq and patents on the basis of Importance of attributes

The model looks like below,

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.913
Model:                            OLS   Adj. R-squared:                  0.912
Method:                 Least Squares   F-statistic:                     1101.
Date:                Mon, 17 Oct 2022   Prob (F-statistic):          1.32e-275
Time:                        12:54:46   Log-Likelihood:                822.21
No. Observations:                 531   AIC:                            -1632.
Df Residuals:                     525   BIC:                            -1607.
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            0.0283      0.009      3.321      0.001       0.012       0.045
capital          0.4778      0.029     16.588      0.000       0.421       0.534
randd            0.1481      0.051      2.918      0.004       0.048       0.248
employment       0.4595      0.025     18.162      0.000       0.410       0.509
sp500            0.0113      0.004      3.151      0.002       0.004       0.018
institutions     0.0021      0.003      0.808      0.419      -0.003       0.007
==============================================================================
Omnibus:                      233.042   Durbin-Watson:                   1.900
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2108.340
Skew:                           1.684   Prob(JB):                         0.00
Kurtosis:                      12.163   Cond. No.                         27.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Now comparing to other model, the R and adj R2 has not been changed much compared to other scenario and we can able to get the best attributes here

Paramas of the Model are as below

```
const           0.028299
capital         0.477768
randd           0.148145
employment      0.459500
sp500           0.011257
institutions    0.002081
dtype: float64
```

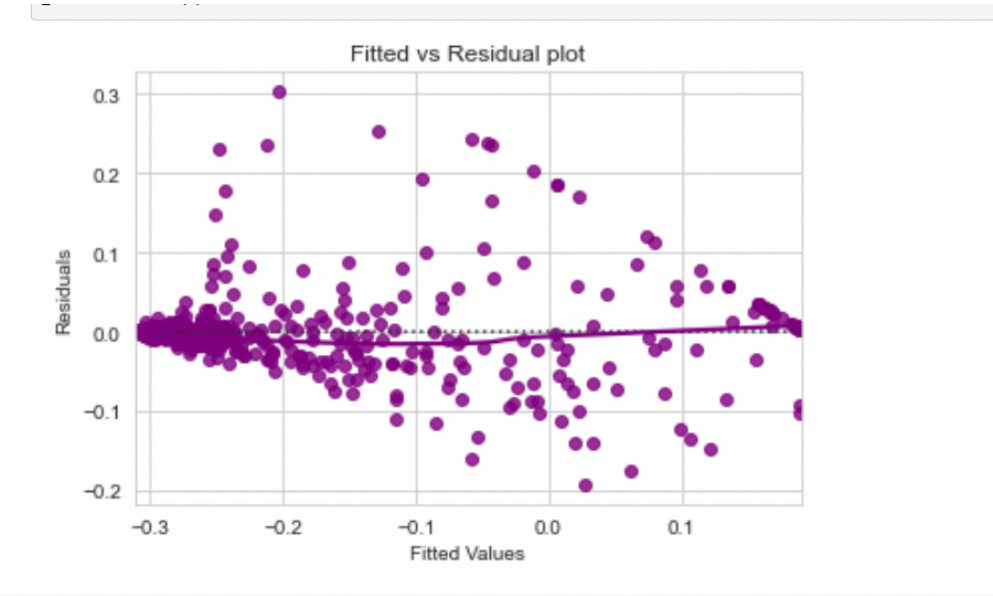
The equation of the model are as follow,

```
price = 0.028299115503080438 + 0.4777681981583373 * ( capital ) +  0.14814498002360377 * ( randd ) +  0.4595002172308
1313 * ( employment ) +  0.011256618008760263 * ( sp500 ) +  0.00208124381198248 * ( institutions )
```

Some Assumption on Linear Regression-

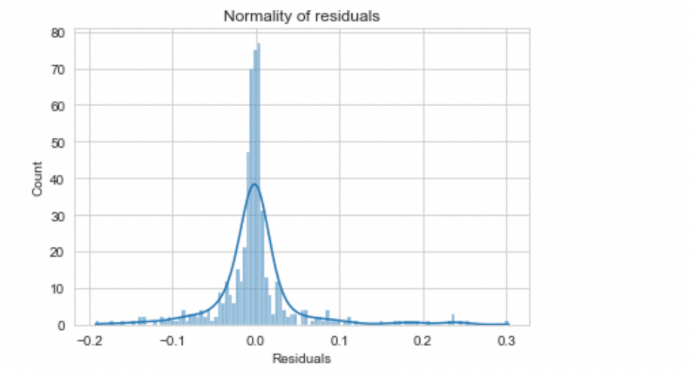
1).Non Linear on Residual

The Data below are Non linear in Nature



2).Normality

The data are normally Distributed



3)Homoscedastic of Data

The data are Homoscedastic in nature

```
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
```

```
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
name = ["F statistic", "p-value"]
test = sms.het_goldfeldquandt(df_pred["Residuals"], X_train_3)
lzip(name, test)
```

```
[('F statistic', 0.7125740182483603), ('p-value', 0.996735494904524)]
```

```
* Since p-value > 0.05 we can say that the residuals are homoscedastic.
```

4)Collinearity

Using VIF, we determined that there is no multicollinearity is present in the data

```
VIF values:

const           14.765009
capital          3.789617
patents          2.581241
randd            3.075643
employment       4.625513
sp500            2.630212
tobinq           1.169457
institutions     1.253192
dtype: float64
```

RMSE of the Model

The RMSE of the Model for Training and testing are very low and we can deploy the Model

```
rmse1 = np.sqrt(mean_squared_error(y_train, df_pred["Fitted Values"]))
rmse1
```

```
0.05143915392808871
```

```
rmse2 = np.sqrt(mean_squared_error(y_test, y_pred))
rmse2
```

```
0.046597073933670144
```

Mean Absolute Error

The MAE of the both training and Testing are also too low

```
mae1 = mean_absolute_error(y_train, df_pred["Fitted Values"])
mae1
```

```
0.026944182520200838
```

```
mae2 = mean_absolute_error(y_test, y_pred)
mae2
```

```
0.02428101464163583
```

R squared and Adjusted R2 are also 91.3% and 91.2%

The Prediction of the Data

price = 0.028299115503080438 + 0.4777681981583373 * (-0.168736) + 0.14814498002360377 * (-0.203250) + 0.45950021723081313 * (-0.227855) + 0.011256618008760263 * (-0.632747) + 0.00208124381198248 * (-1.461405)

Price= -0.19729159975652844

The Data has been predicted correctly using trained value on the Tested Value

	const	capital	patents	randd	employment	sp500	tobinq	value	institutions	y_pred
281	1.0	-0.168736	-0.265767	-0.203250	-0.227855	-0.632747	-5.815161e-01	-0.297473	-1.461405	-0.197292
682	1.0	-0.217638	-0.018842	-0.165127	-0.142391	1.580410	-4.302729e-01	-0.220593	1.448435	-0.144768
12	1.0	-0.304349	-0.255479	-0.212001	-0.326001	-0.632747	1.012388e+00	-0.373694	-1.475248	-0.308506
306	1.0	-0.283355	-0.224613	-0.202938	-0.302949	-0.632747	-5.261830e-01	-0.374972	-0.459158	-0.284426
601	1.0	0.097912	0.014596	-0.053974	0.221035	1.580410	4.185673e-02	0.318022	0.438804	0.187351
...	...	...	...	...	...	...	...	...	...	...
574	1.0	-0.298520	-0.234901	-0.210281	-0.299576	-0.632747	-3.174265e-01	-0.371529	0.672754	-0.288854
654	1.0	-0.292910	-0.173170	-0.209724	-0.266545	-0.632747	-1.619858e-01	-0.327705	0.438804	-0.271400
258	1.0	-0.303616	-0.245190	-0.215672	-0.322421	-0.632747	4.424334e-01	-0.375826	-1.131475	-0.306339
629	1.0	-0.302909	-0.245190	-0.220752	-0.323137	-0.632747	-3.362317e-01	-0.385223	-1.498782	-0.307848
288	1.0	-0.289847	-0.204036	-0.160061	-0.285140	-0.632747	1.874110e-15	-0.360705	-0.421781	-0.272915

The Data are scaled in Nature, So the Value is replicating the Negative Value

Hence by Rescaling we can get the Original Values

Scenario 2

In below scenario, the RMSE and MSE are higher compared to Scenario 1

The R and Adj R2 reduced drastically to the 0.859 and 0.857. The Scenario can be Rejected We had dropped education from the data instead of Patents.So education cannot be dropped at any scenario

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.859
Model:                            OLS   Adj. R-squared:                  0.857
Method:                 Least Squares   F-statistic:                     638.3
Date:                Thu, 13 Oct 2022   Prob (F-statistic):          1.96e-220
Time:                        19:51:11   Log-Likelihood:                 693.63
No. Observations:                 531   AIC:                            -1375.
Df Residuals:                     525   BIC:                            -1350.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0554      0.011      5.142      0.000       0.034       0.077
capital        0.7940      0.029     27.231      0.000       0.737       0.851
patents        0.0565      0.043      1.308      0.192      -0.028       0.141
randd          0.3406      0.082      4.160      0.000       0.180       0.501
sp500          0.0321      0.004      7.448      0.000       0.024       0.041
institutions   0.0044      0.003      1.349      0.178      -0.002       0.011
==============================================================================
Omnibus:                       58.809   Durbin-Watson:                   2.072
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              369.522
Skew:                           0.177   Prob(JB):                     5.75e-81
Kurtosis:                       7.071   Cond. No.                         36.5
==============================================================================
```

```
rmse1 = np.sqrt(mean_squared_error(y_train, df_pred["Fitted Values"]))
rmse1
```

```
0.06553279591419017
```

```
rmse2 = np.sqrt(mean_squared_error(y_test, y_pred))
rmse2
```

```
0.06282285147910023
```

```
mae1 = mean_absolute_error(y_train, df_pred["Fitted Values"])
mae1
```

```
0.04073158714238751
```

```
mae2 = mean_absolute_error(y_test, y_pred)
mae2
```

```
0.036687154417431214
```

Scenario 3

In below scenario, the RMSE and MSE are higher compared to Scenario 1

The R and Adj R2 reduced drastically to the 0.859 and 0.857. The Scenario can be Rejected we had dropped all expect the capital,randd,sp500 and Institutions

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.858
Model:                            OLS   Adj. R-squared:                  0.857
Method:                 Least Squares   F-statistic:                     796.4
Date:                Thu, 13 Oct 2022   Prob (F-statistic):          1.53e-221
Time:                        17:54:54   Log-Likelihood:                692.76
No. Observations:                 531   AIC:                            -1376.
Df Residuals:                     526   BIC:                            -1354.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            0.0573      0.011      5.374      0.000       0.036       0.078
capital          0.7962      0.029     27.334      0.000       0.739       0.853
randd            0.4105      0.062      6.617      0.000       0.289       0.532
sp500            0.0322      0.004      7.466      0.000       0.024       0.041
institutions     0.0046      0.003      1.403      0.161      -0.002       0.011
==============================================================================
Omnibus:                       58.500   Durbin-Watson:                   2.068
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              373.067
Skew:                           0.158   Prob(JB):                     9.76e-82
Kurtosis:                       7.094   Cond. No.                         26.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
rmse1 = np.sqrt(mean_squared_error(y_train, df_pred["Fitted Values"]))
rmse1
```

```
0.06563944839658423
```

```
rmse2 = np.sqrt(mean_squared_error(y_test, y_pred))
rmse2
```

```
0.06253067730134963
```

```
mae1 = mean_absolute_error(y_train, df_pred["Fitted Values"])
mae1
```

```
0.04085328471712768
```

```
mae2 = mean_absolute_error(y_test, y_pred)
mae2
```

```
0.03656376718804175
```

So from the Over all basis, Scenario 1 better compared other scenarios because of the Higher R2 and adj R2 values in data and lesser error in RMSE and MAE

So the Scenario 1 is best for the Predictions and the 5 attributes that contribute much for the data are as follow

```
X_train_2.columns
```

```
Index(['const', 'capital', 'randd', 'employment', 'sp500', 'institutions'], dtype='object')
```

Capital ,randd, employment,SP 500 and Institutions are the top 5 attributes

2.4) Inference: Based on these predictions, what are the insights and recommendations? (6 marks)

- The capital such as Net Stock of Property,plant and Equipment which plays major in contributing the Sales to the Firm.The Major aspect as an Investment firm is to check the company with the Higher Capital which in turn make an higher Sales
- The Second is most important aspect is to check the Employment of the Firm, Whenever the employment got Increased the company make better sales. As an investor is to check the employment in the firm of the Company and to make Better decision while investing in the firm
- The R&D plays the role of higher sales. So while Investing we should check how much the Firm is keen on Investing the R & D. This Parameter makes small edge over other Investment Firm
- The Institution such Proportion of Stock owned by Institutions. The Institutions will make huge Investment compared to the retail Investor. So Being an Firm We need to check how much institution contributing for Stock price will make an added advantage while Investing
- In terms of SP 500, the member of Member of SP 500 has higher chance of better sales. So our most important aspect here is to check the SP500 company
- Other Attributes such Patents,Dobinq, value etc plays a role but less than what is compared in above in the Model. So we can still concentrate those factor also

Problem 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.

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 population's sampling unit, the case number, and the vehicle number. Within each year, use this to uniquely identify the vehicle.

	weight	frontal	ageOFocc	yearacc	yearVeh	deploy	injSeverity
count	11217.000000	11217.000000	11217.000000	11217.000000	11217.000000	11217.000000	11140.000000
mean	431.405309	0.644022	37.427654	2001.103236	1994.177944	0.389141	1.825583
std	1406.202941	0.478830	18.192429	1.056805	5.658704	0.487577	1.378535
min	0.000000	0.000000	16.000000	1997.000000	1953.000000	0.000000	0.000000
25%	28.292000	0.000000	22.000000	2001.000000	1991.000000	0.000000	1.000000
50%	82.195000	1.000000	33.000000	2001.000000	1995.000000	0.000000	2.000000
75%	324.056000	1.000000	48.000000	2002.000000	1999.000000	1.000000	3.000000
max	31694.040000	1.000000	97.000000	2002.000000	2003.000000	1.000000	5.000000

4	55+	13.374	Not_Survived	none	none	1	m	23	1997	1986.0	unavail	driver	0	4.0	4:58:1

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. (8 marks)

The head of the Data are as follow

The shape of the data after removing unnamed:0 as

(11217,15)

```
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dvcat        11217 non-null  object
 1   weight       11217 non-null  float64
 2   Survived     11217 non-null  object
 3   airbag       11217 non-null  object
 4   seatbelt     11217 non-null  object
 5   frontal      11217 non-null  int64
 6   sex          11217 non-null  object
 7   ageOFocc     11217 non-null  int64
 8   yearacc      11217 non-null  int64
 9   yearVeh      11217 non-null  float64
 10  abcat        11217 non-null  object
 11  occRole      11217 non-null  object
 12  deploy       11217 non-null  int64
 13  injSeverity  11140 non-null  float64
 14  caseid       11217 non-null  object
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

From the Info of the Data we see clearly that there is some missing Value in injseverity . The total object is 8,where as float and Integer is totally 7. So we need to check the Missing data, Anomalies and Bad value in the Data and it should be treated

For Treating Missing data, Anomalies and Bad value in the Data, info and describe of the should be Verified treating those Value

In above weight is 0 which is kind of Anomalies which should be treated at first, but we can see lot of weight ranging from 0 to 1 which is higher than that of 0 so I had decided not to treat those Anomalies unless and until if there is issue with my model

The Next step is to treat the Missing Value present in the injseverity

```
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
```

The total Injseverity missing in Data is 77,This 77 should be treated for the missing value

Here the size of the model is 11217, so we can drop those 77 data which data become 10400 which is still holding good for prediction

There is no bad value present in the Data and total of two duplicated row and the row got dropped

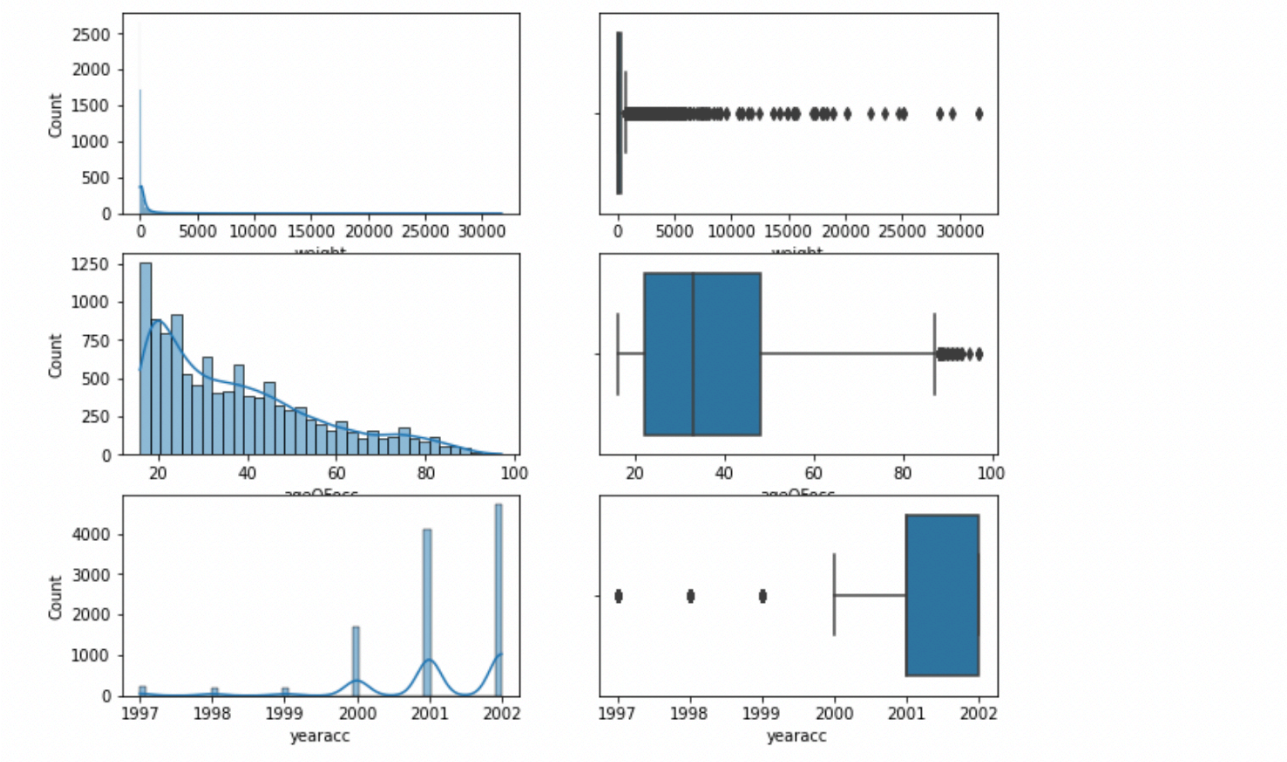
Hence above the Null check condition is verified

Data Visualisation

Univariate Analysis

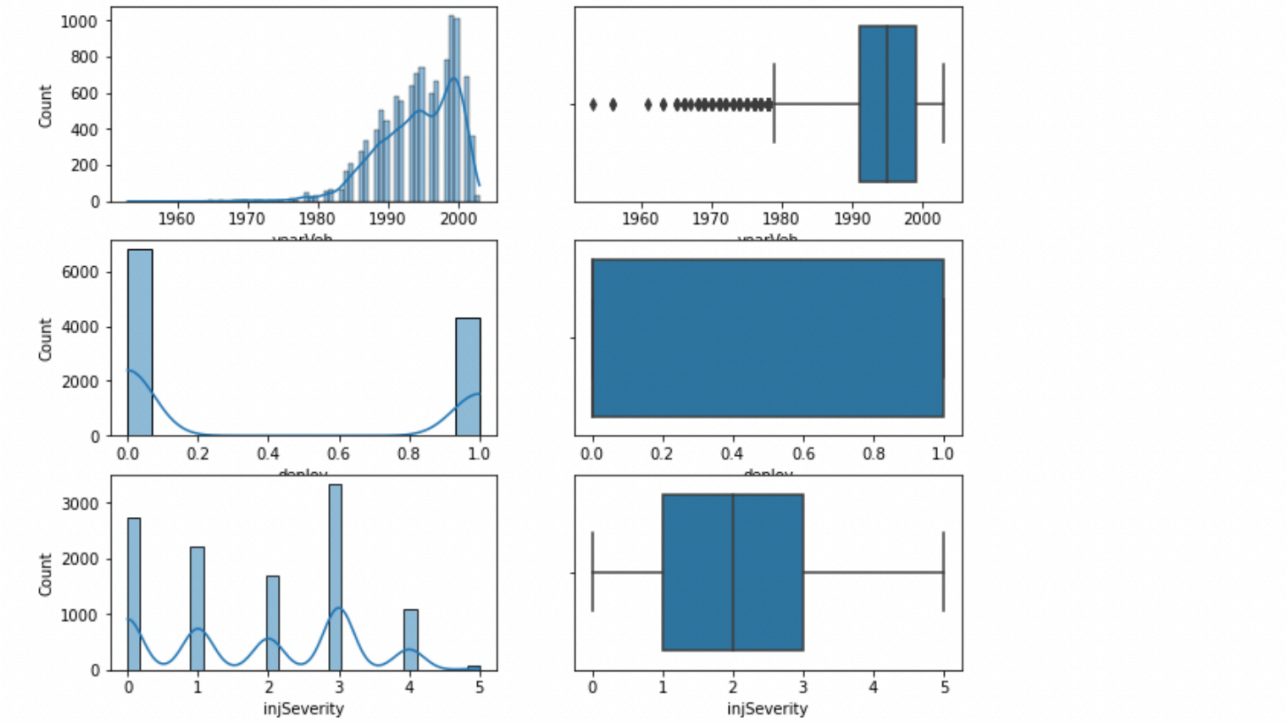
Univariate data requires to analyze each variable separately. Uni means one, so in other words the data has only one variable.In Univariate Analysis, we can use Histogram and Box plot for Numerical type where as countplot for Categorical type

The Histogram and Box Plot Analysis of the eight numeric data are given below



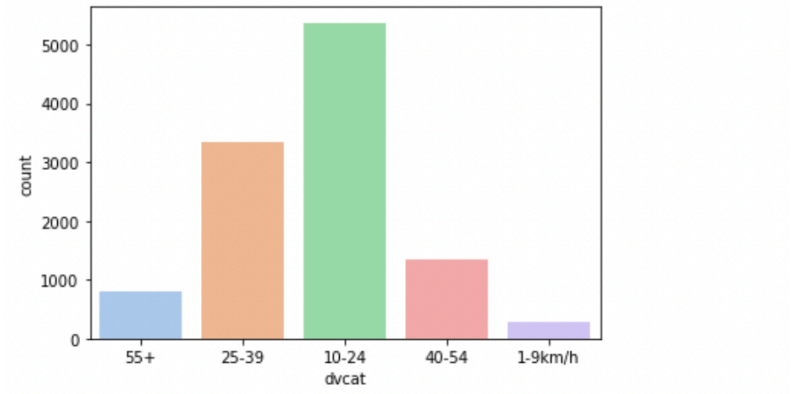
The Six Numerical Data is presented in the form of Histogram and Boxplot and from the above observation we can clearly see that there is lots of Outlier present in the Data and also Histogram are not much Normally Distributed. Hence Treatment of Outliers and Standardisation is required for the above data.

Categorical Variable

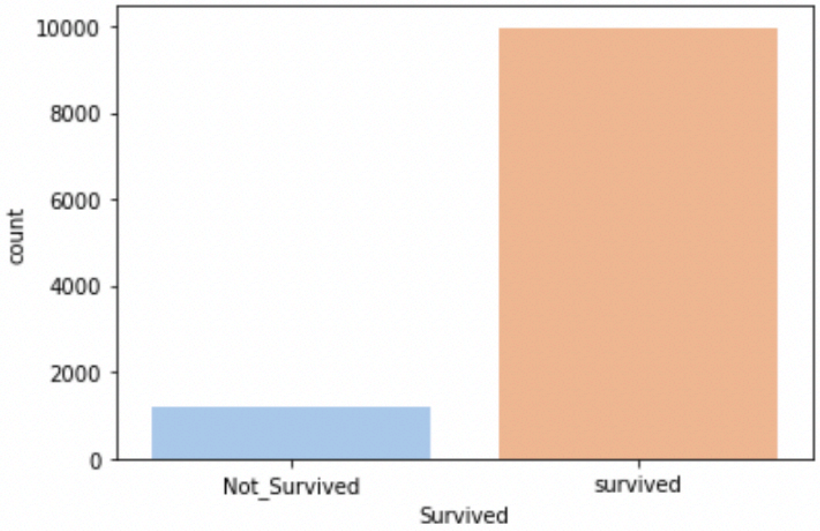


In Below are analyse of the Categorical in the data with the value counts. Hence the they are attached below

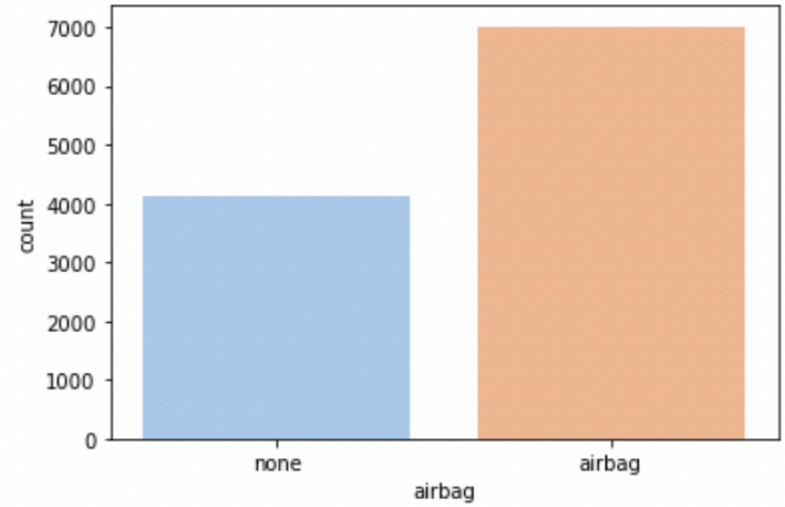
```
10-24       5370
25-39       3346
40-54       1338
55+          808
1-9km/h      276
Name: dvcat, dtype: int64
```



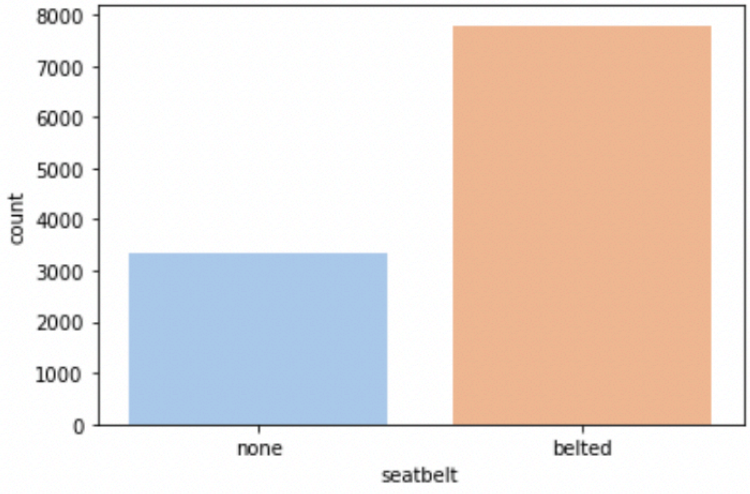
```
survived        9958
Not_Survived    1180
Name: Survived, dtype: int64
```



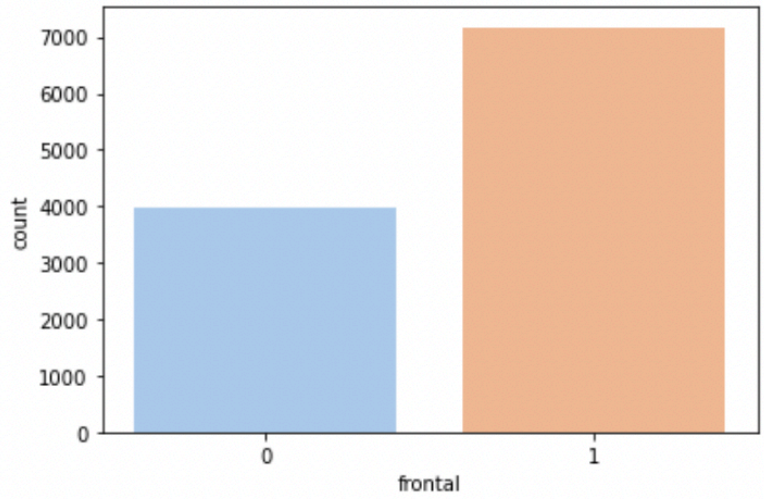

```
airbag    7012
none      4126
Name: airbag, dtype: int64
```



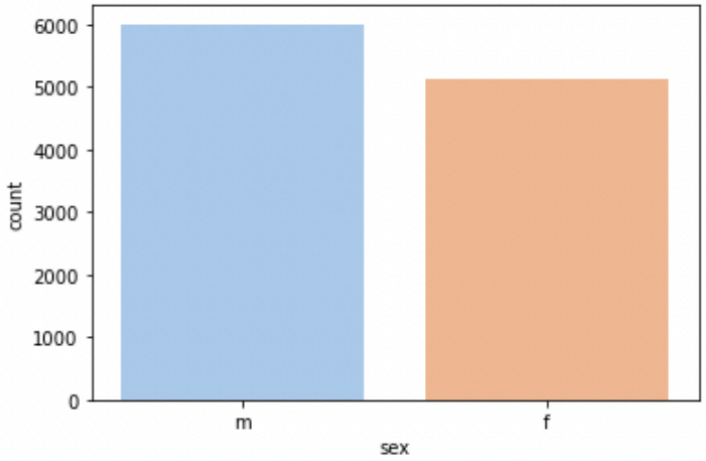

```
belted    7788
none      3350
Name: seatbelt, dtype: int64
```



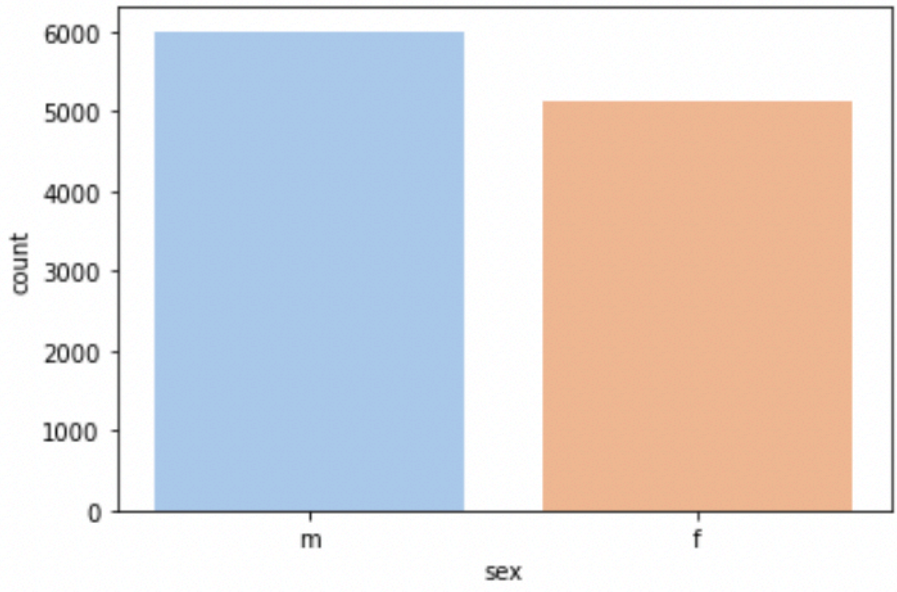
```
1    7171
0    3967
Name: frontal, dtype: int64
```



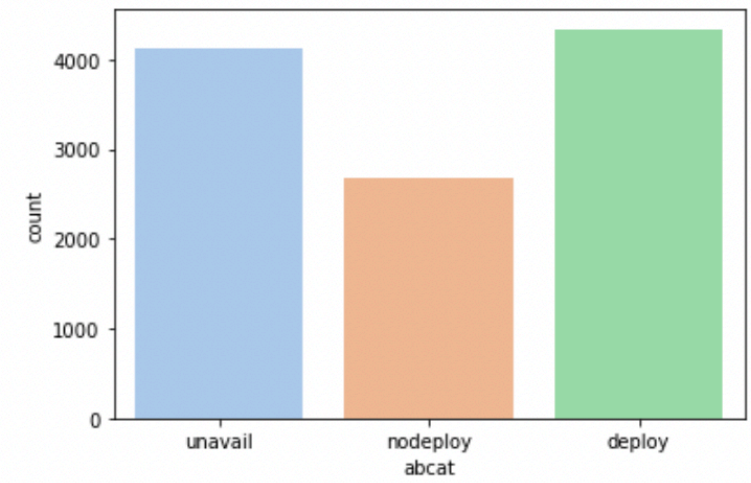

```
m    6004
f    5134
Name: sex, dtype: int64
```



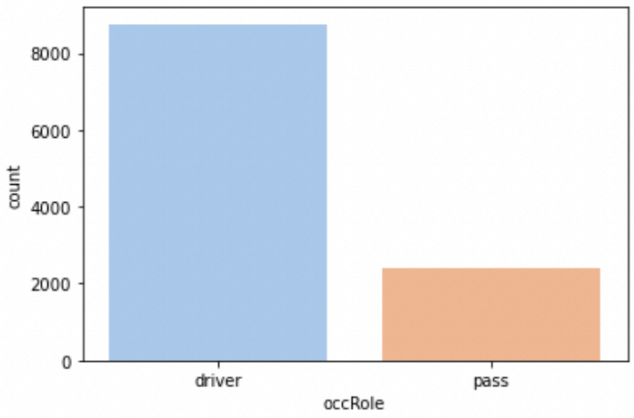

```
m    6004
f    5134
Name: sex, dtype: int64
```

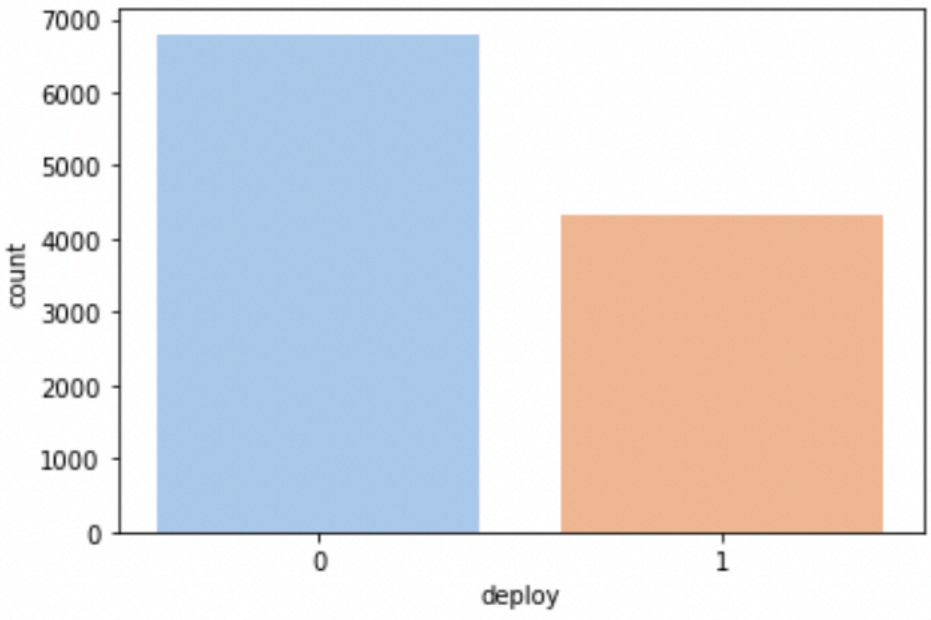
```
deploy      4340
unavail     4126
nodeploy    2672
Name: abcat, dtype: int64
```



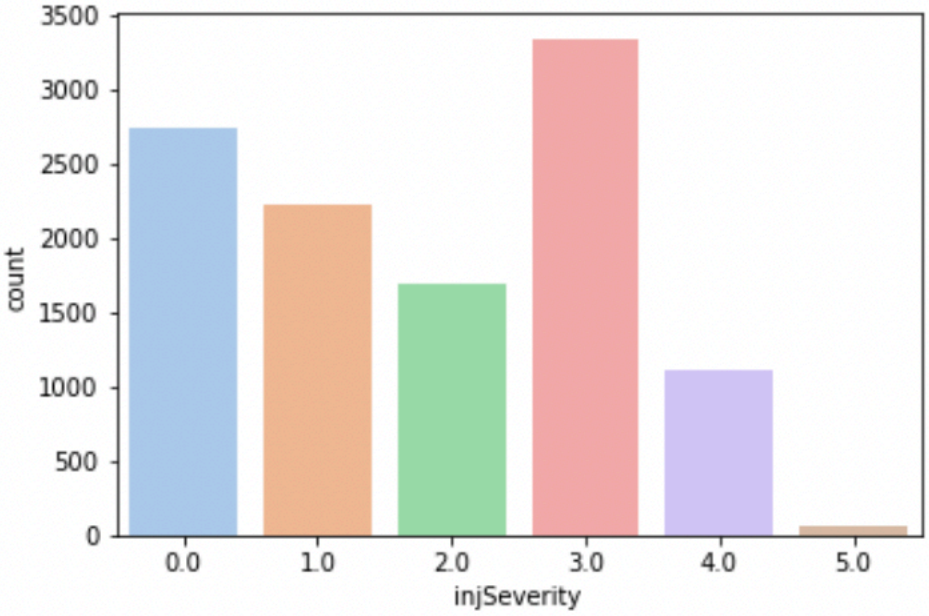

```
driver    8753
pass      2385
Name: occRole, dtype: int64
```




```
0    6798
1    4340
Name: deploy, dtype: int64
```



```
3.0    3336
0.0    2733
1.0    2218
2.0    1682
4.0    1101
5.0      68
Name: injSeverity, dtype: int64
```



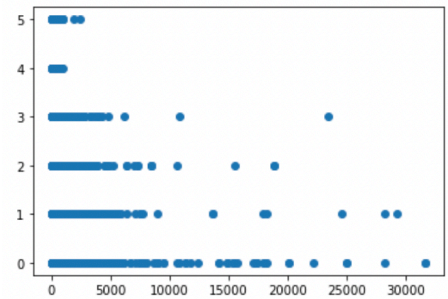
Bivariate Analysis

In Bivariate Analysis, there are two variables wherein the analysis is related to cause and the relationship between the two variables.Here the Numeric and Numeric relation are established in the relationship using Pairplot/scatter plot whereas Categorical and Categorical are established with Countplot with Hue

Scatter Plot

```
plt.scatter(df['weight'],df['injSeverity'])
```

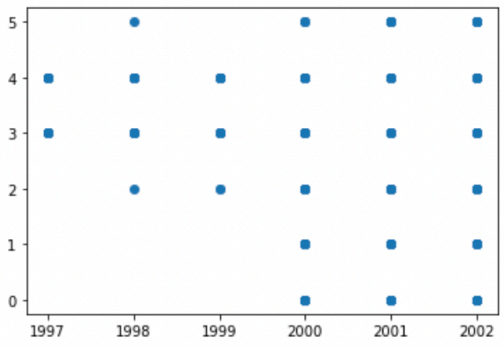
```
<matplotlib.collections.PathCollection at 0x7f819530bd60>
```



In terms between the InjSeverity and Weight,Higher the Weight the injseverity is low

```
plt.scatter(df['yearacc'],df['injSeverity'])
```

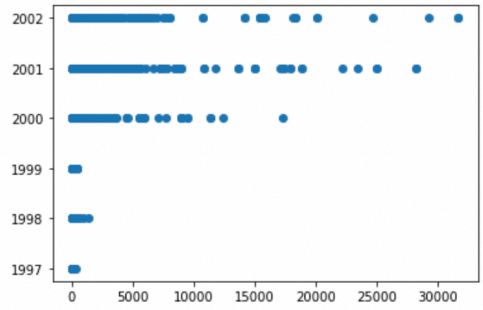
```
<matplotlib.collections.PathCollection at 0x7f8180e8c6d0>
```



In between the ‘yearacc’ and ‘Injseverity’ ,The Injseverity got reduced when the when the year got Increased which we can identify that the Introduction of many safety measure got reduced the Injseverity

```
plt.scatter(df['weight'],df['yearacc'])
```

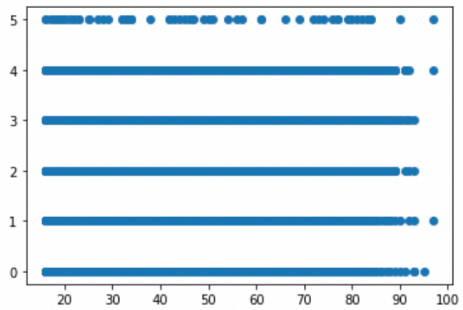
```
<matplotlib.collections.PathCollection at 0x7f819524d730>
```



Even interms of weight also we can see that weight got Increased as Year Increases

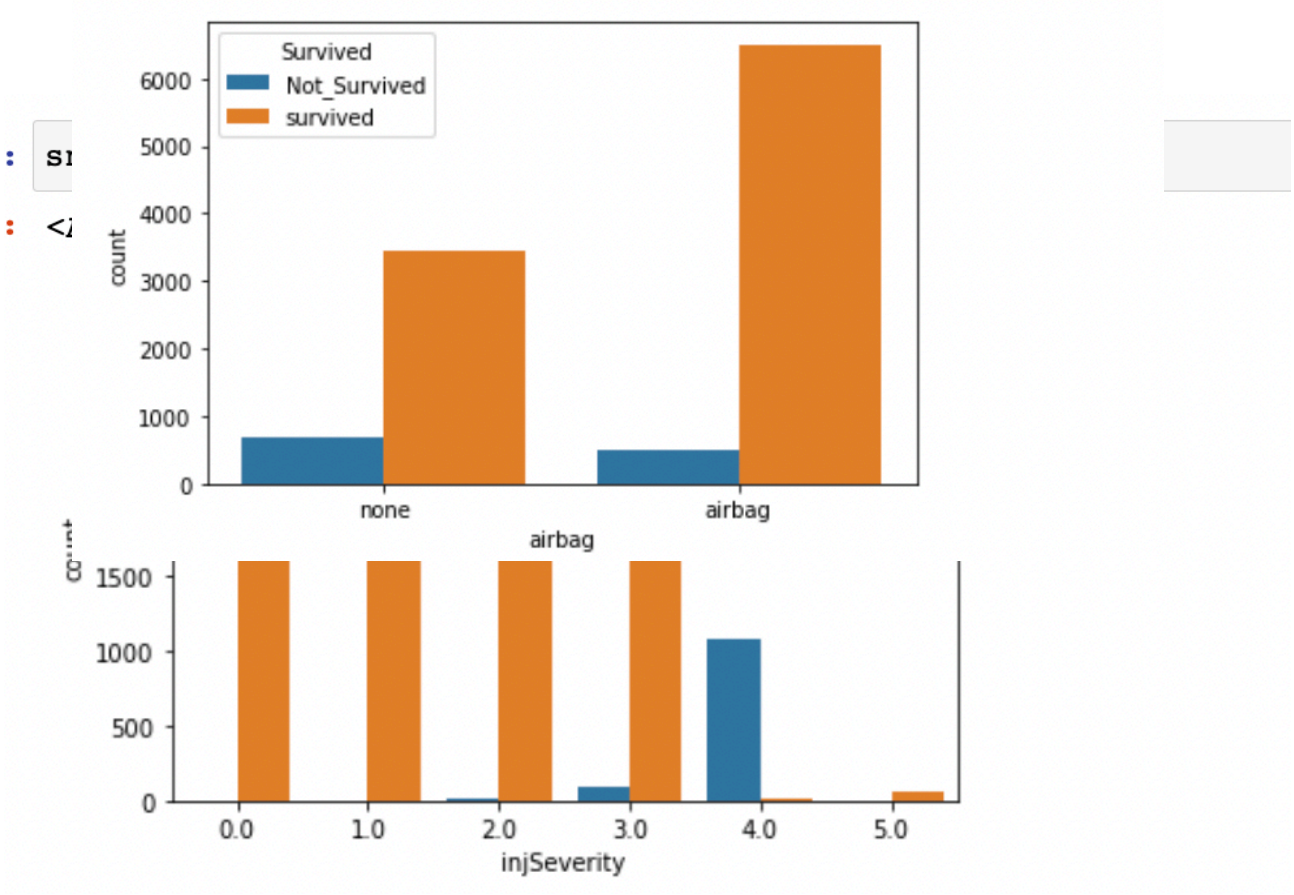
```
plt.scatter(df['ageOFocc'],df['injSeverity'])
```

```
<matplotlib.collections.PathCollection at 0x7f8171e69640>
```

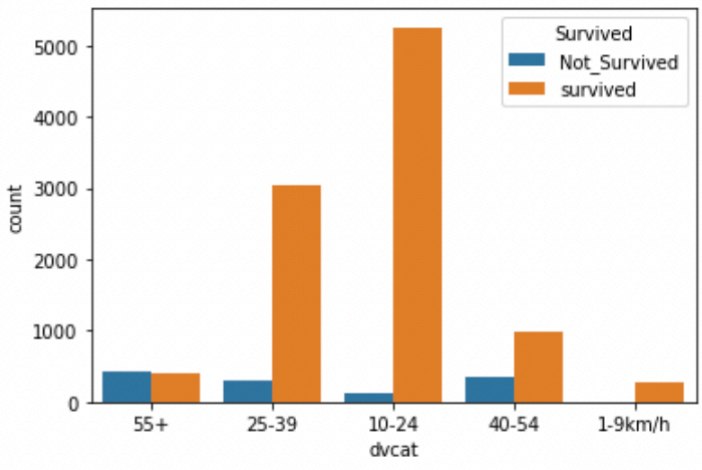


In term of ageOFocc and injseverity we can see that there is kind of no relationship

Comparing Independent Variable and Dependent Variable using Bi_variate Analysis



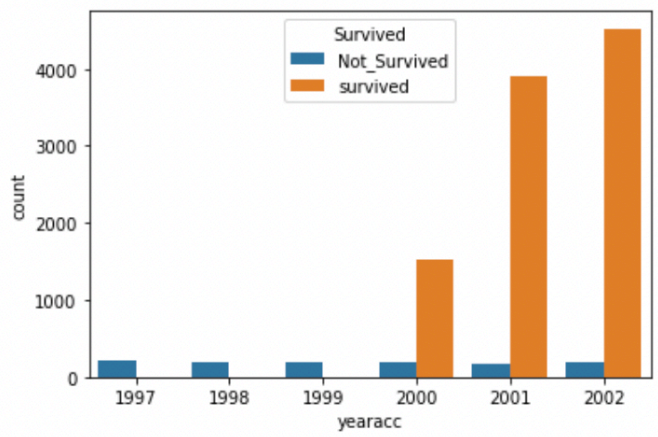
When Injseverity is low, the People got Survived and when it is above 3 there is kind of some causality



When speed is low, there is no causality but when it is above there is kind of Non Survival,When above 55+ there there is less 50% chance for Survival

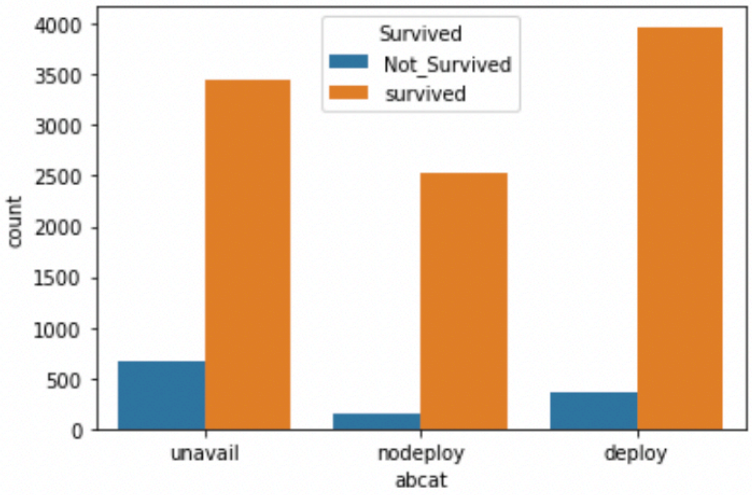
Whe with the Airbag, the Survival rate is Higher compared to Non Survival without the Airbag

```
: sns.countplot(hue='Survived',x ='yearacc', data=df,dodge='bool')
: <AxesSubplot:xlabel='yearacc', ylabel='count'>
```



Initially in year such as 1997, there is non survival because the reason might be such as Technology, Awarness etc but later in the year on above 2000 there is Kind of Survival Existed but still there is causalities

```
: sns.countplot(x='abcat',hue ='Survived', data=df,dodge='bool')
: <AxesSubplot:xlabel='abcat', ylabel='count'>
```



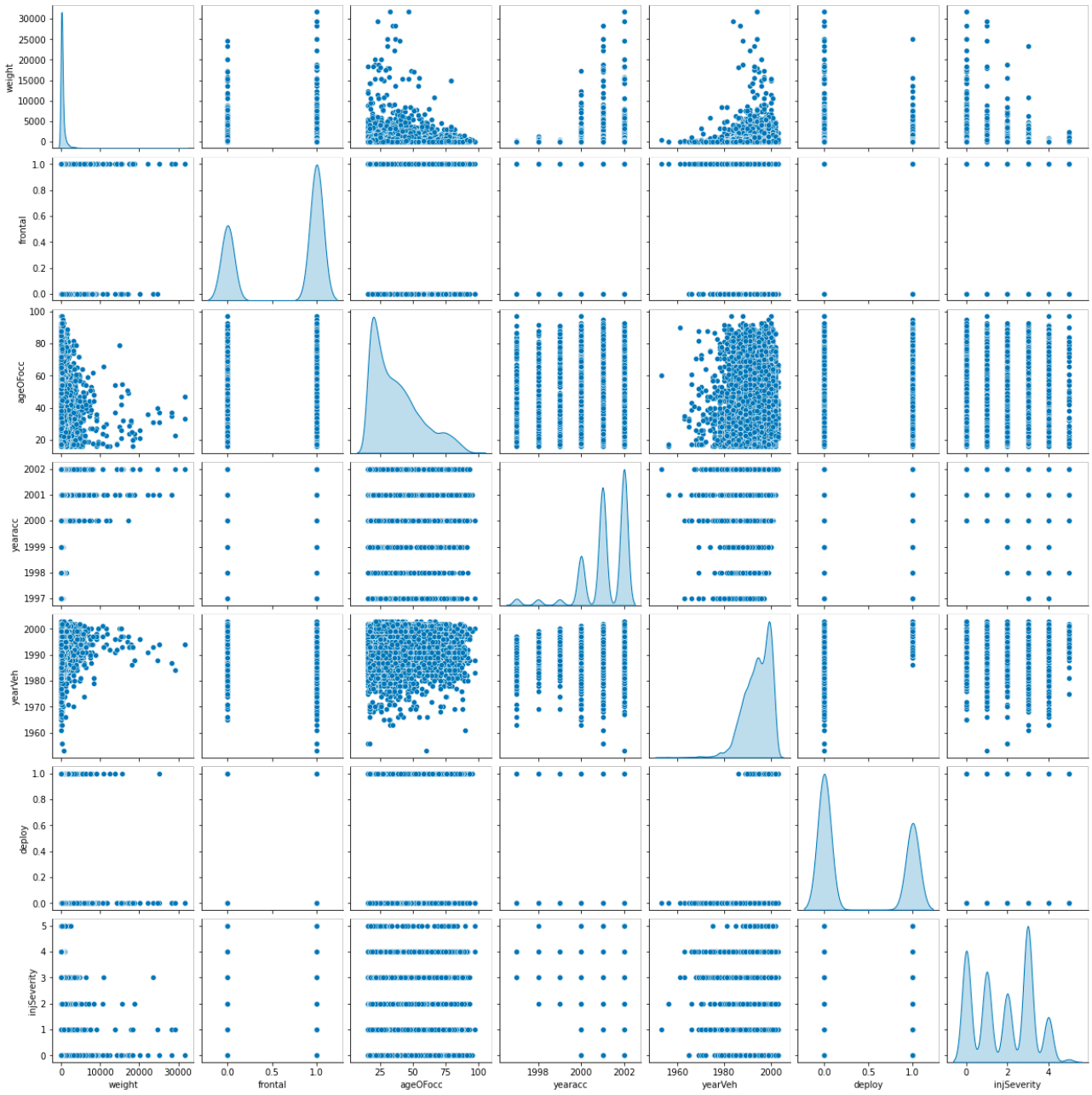
If the Airbag is Unavailed or Nor Deployed then the chance of Causality is higher, When deployed the Survival Rate is higher compared to other scenarios

Multivariate Analysis

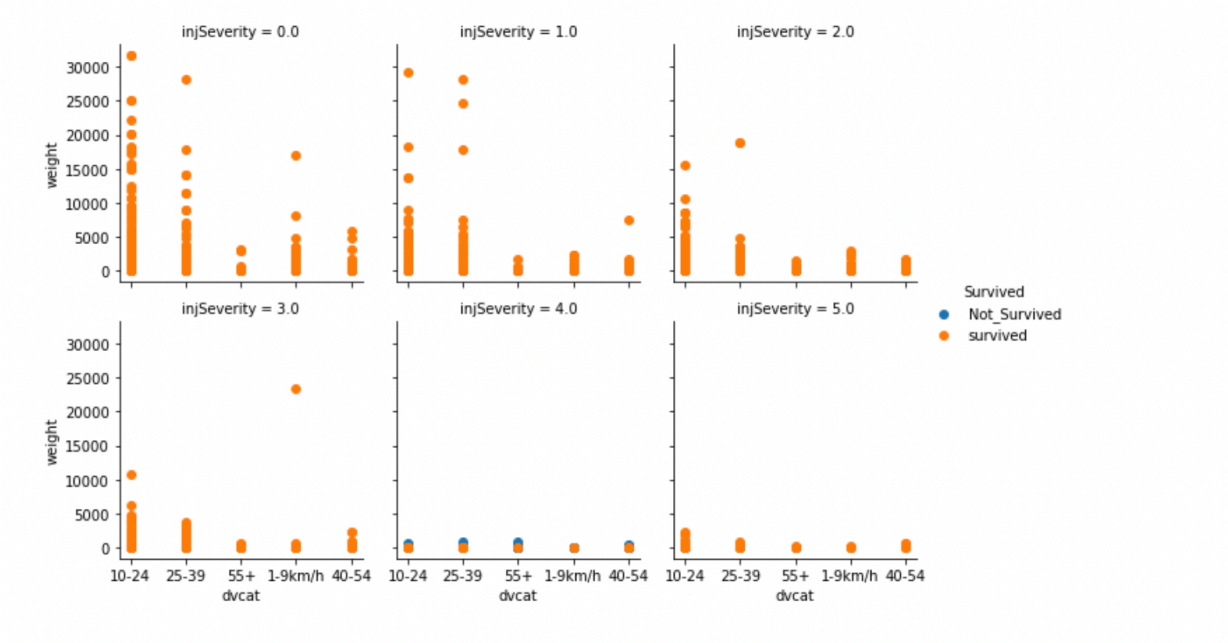
Multivariate descriptive displays or plots are designed to reveal the relationship among several variables simulataneously.. As was the case when examining relationships among pairs of variables, there are several basic characteristics of the relationship among sets of variables that are of interest. Some of the Interesting Multivariate Analysis are pair plot,Heat Map, Facet etc

Pairwise Analysis

A pairs plot allows us to see both distribution of single variables and relationships between two variable.The Pairwise plots are Mentioned below which determines the Overall relationship between the Dependent Variable and all other Independent variable. From this we Can identify microlevel and we can establish some kind of Relation which is explained below

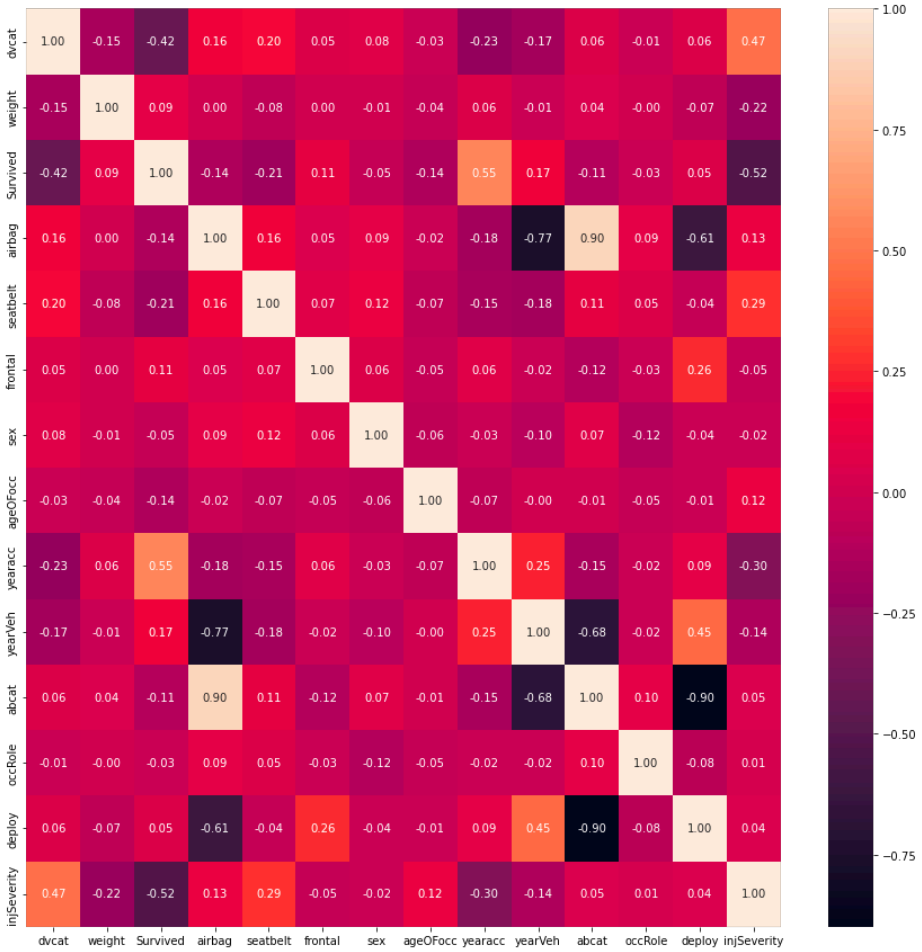


Facetgrid- facet_grid() forms a matrix of panels defined by row and column faceting variables. It is most useful when you have two discrete variables, and all combinations of the variables exist in the data



Correlation in Data

The Correlation in the following data can be explained in the form of the Pictorial in Heat Map



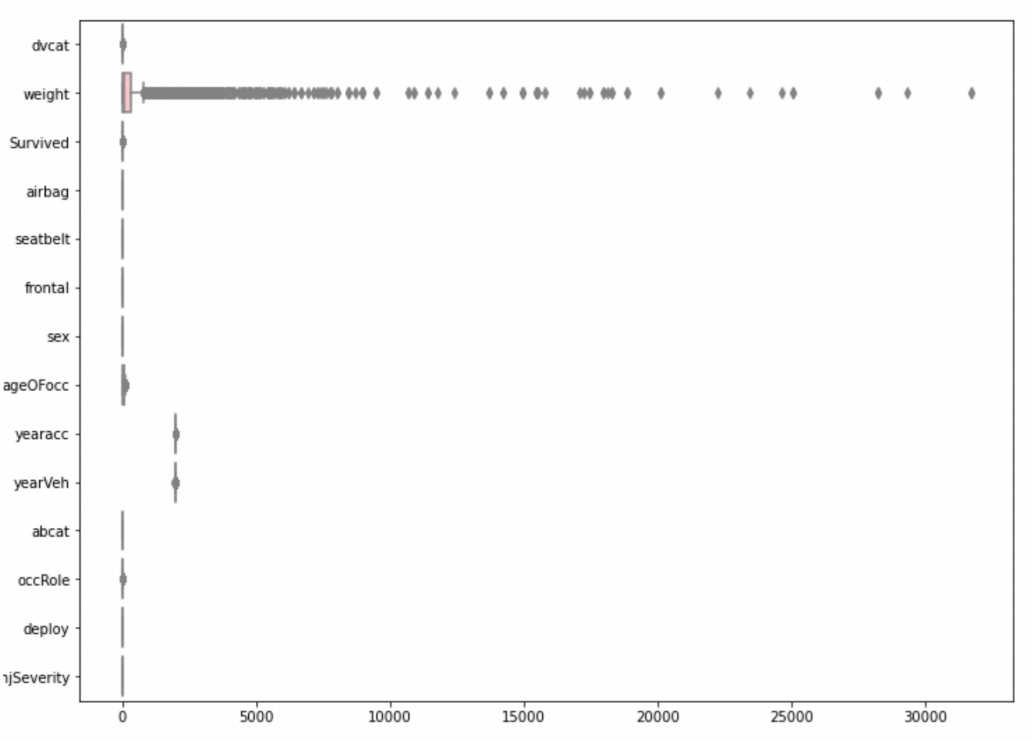
	dvcat	weight	Survived	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat	occRole	deploy	injSeverity
dvcat	1.000000	-0.145575	-0.416078	0.160148	0.204397	0.050982	0.076014	-0.034728	-0.233547	-0.165226	0.057649	-0.012194	0.055551	0.473674
weight	-0.145575	1.000000	0.091673	0.003180	-0.078827	0.000794	-0.006862	-0.039823	0.056946	-0.014603	0.038530	-0.000266	-0.065716	-0.221584
Survived	-0.416078	0.091673	1.000000	-0.140072	-0.206798	0.108268	-0.046769	-0.135930	0.550338	0.165362	-0.108330	-0.025832	0.054912	-0.521198
airbag	0.160148	0.003180	-0.140072	1.000000	0.156491	0.049520	0.093189	-0.024614	-0.182592	-0.766293	0.896972	0.087692	-0.612912	0.125351
seatbelt	0.204397	-0.078827	-0.206798	0.156491	1.000000	0.066298	0.117499	-0.068978	-0.148935	-0.179173	0.111265	0.050903	-0.043899	0.285491
frontal	0.050982	0.000794	0.108268	0.049520	0.066298	1.000000	0.055830	-0.049910	0.059096	-0.023695	-0.117924	-0.034525	0.259806	-0.054227
sex	0.076014	-0.006862	-0.046769	0.093189	0.117499	0.055830	1.000000	-0.061614	-0.026275	-0.097348	0.072399	-0.116638	-0.037119	-0.021449
ageOFocc	-0.034728	-0.039823	-0.135930	-0.024614	-0.068978	-0.049910	-0.061614	1.000000	-0.071792	-0.002020	-0.007856	-0.051234	-0.010334	0.124803
yearacc	-0.233547	0.056946	0.550338	-0.182592	-0.148935	0.059096	-0.026275	-0.071792	1.000000	0.248768	-0.152308	-0.020859	0.091409	-0.303425
yearVeh	-0.165226	-0.014603	0.165362	-0.766293	-0.179173	-0.023695	-0.097348	-0.002020	0.248768	1.000000	-0.678237	-0.021189	0.453393	-0.140155
abcat	0.057649	0.038530	-0.108330	0.896972	0.111265	-0.117924	0.072399	-0.007856	-0.152308	-0.678237	1.000000	0.095625	-0.899081	0.048474
occRole	-0.012194	-0.000266	-0.025832	0.087692	0.050903	-0.034525	-0.116638	-0.051234	-0.020859	-0.021189	0.095625	1.000000	-0.084074	0.013630
deploy	0.055551	-0.065716	0.054912	-0.612912	-0.043899	0.259806	-0.037119	-0.010334	0.091409	0.453393	-0.899081	-0.084074	1.000000	0.037492
injSeverity	0.473674	-0.221584	-0.521198	0.125351	0.285491	-0.054227	-0.021449	0.124803	-0.303425	-0.140155	0.048474	0.013630	0.037492	1.000000

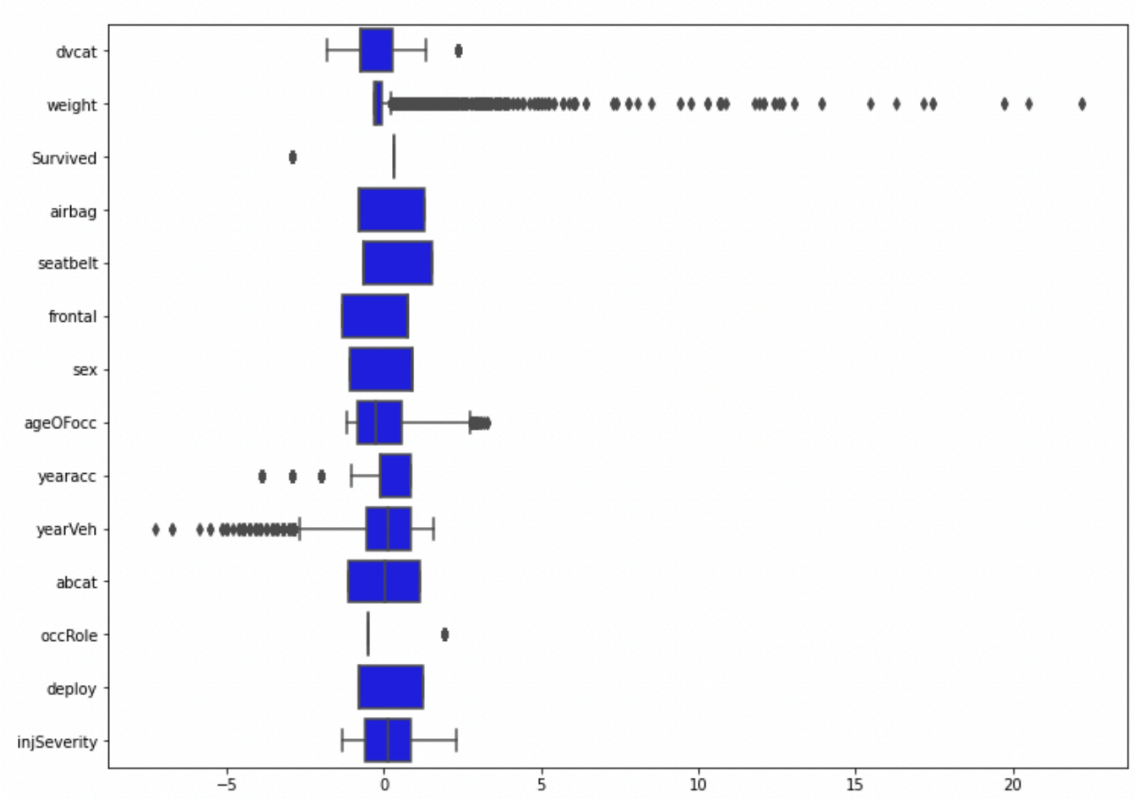
Correlation measures the relationship between two variables. We mentioned that a function has a purpose to predict a value. We can say also say that a function uses the relationship between two variables for prediction.In above Pictorical Representation will represent more Above about the Correlation

Detecting the Outlier in the Data

Along With those Analysis one of important aspect is the Detecting the Outlier in the Data

We Can see in below that only weight has Outlier while rest of them don't have any kind of Outlier Its will clearly visible after doing the Scaling of the Data





We can Cap the Outlier only in Weight and Year Vech while rest are with the lower values.Hence there is no requirement of detecting outlier in other Attributes unless and until it is essential

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). (8 marks)

For Encoding the data present in the variable we need to first Identify the total categorical variable present in the Data. In the below data we can see there are total of seven Category available where as the rest of the data present are in int or Float.

The second is to identify how to encode those categorical variable into numerical, All the data mentioned below are ordinal, so we can encode those data point directly instead of using the one hot Encoder

Before Converting the categorical into the Numerical Variable

```
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dvcat        11138 non-null  object
 1   weight       11138 non-null  float64
 2   Survived     11138 non-null  object
 3   airbag       11138 non-null  object
 4   seatbelt     11138 non-null  object
 5   frontal      11138 non-null  int64
 6   sex          11138 non-null  object
 7   ageOFocc     11138 non-null  int64
 8   yearacc      11138 non-null  int64
 9   yearVeh      11138 non-null  float64
 10  abcat        11138 non-null  object
 11  occRole      11138 non-null  object
 12  deploy       11138 non-null  int64
 13  injSeverity  11138 non-null  float64
dtypes: float64(3), int64(4), object(7)
memory usage: 1.5+ MB
```

After Converting those Categorical into the numerical Variable

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11138 entries, 0 to 11216
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dvcat        11138 non-null  int8
 1   weight       11138 non-null  float64
 2   Survived     11138 non-null  int8
 3   airbag       11138 non-null  int8
 4   seatbelt     11138 non-null  int8
 5   frontal      11138 non-null  int64
 6   sex          11138 non-null  int8
 7   ageOFocc     11138 non-null  int64
 8   yearacc      11138 non-null  int64
 9   yearVeh      11138 non-null  float64
 10  abcat        11138 non-null  int8
 11  occRole      11138 non-null  int8
 12  deploy       11138 non-null  int64
 13  injSeverity  11138 non-null  float64
dtypes: float64(3), int64(4), int8(7)
memory usage: 1.0 MB
```

Before Splitting the Data, the most essential thing is to Segregate the Independent and Dependent Variable in the Data

```
#Copy all the predictor variables into X dataframe
X = df_1.drop('Survived', axis=1)

# Copy target into the y dataframe.
y = df_1['Survived']
```

Splitting of data

Logistics Regression

```
 #Split X and y into training and test set in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1)
```

In Logistic Regression, We had Segregated the X and y as X_train,X_test,y_train and y_test with the test size of 0.30 and the random state as 1

Linear Discriminant Analysis

```
X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(X, y, test_size=0.30 , random_state=1)
```

In Linear Discriminant Analysis, We had Segregated the X and y as X_train_1,X_test_1,y_train_1 and y_test_1 with the test size of 0.30 and the random state as 1

Model Building on the Data

Logistics Regression

```
model = LogisticRegression(solver='newton-cg',max_iter=2500,penalty='none',verbose=True,n_jobs=2)
model.fit(X_train, y_train)
```

```
[Parallel(n_jobs=2)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done   1 out of   1 | elapsed:    0.8s finished
```

```
LogisticRegression(max_iter=2500, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

In Logistic Regression, We had Build the model using the solver,max_iter,verbose etc to Increase the Accuracy of the Model.We had trained both X and y variable in the Data

```
ytrain_predict = model.predict(X_train)
ytest_predict = model.predict(X_test)
```

```
ytest_predict
```

```
array([1, 1, 0, ..., 0, 1, 1])
```

```
ytrain_predict
```

```
array([1, 1, 1, ..., 1, 1, 1])
```

```
y_test
```

```
2047     1
6536     1
7905     0
6753     1
5770     1
        ..
8264     1
10518    1
487      0
9737     1
10698    1
Name: Survived, Length: 3342, dtype: int64
```

```
ytest_predict_prob=model.predict_proba(X_test)
pd.DataFrame(ytest_predict_prob).head()
```

	0	1
0	1.196354e-01	0.880365
1	7.550237e-03	0.992450
2	7.848651e-01	0.215135
3	1.705868e-06	0.999998
4	1.367998e-08	1.000000

In Above We had predicted for both y train_predict and y test_predict and also interns of Probability also

Linear Discriminant Analysis

```
clfLDA = LinearDiscriminantAnalysis()
clfLDA.fit(X_train_1, y_train_1)
```

```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

In LDA, We had just trained the Data in both X and y variable in the Data and not made any complication in Building the LDA Model

```
ytrain_predict_1 = clfLDA.predict(X_train_1)
ytest_predict_1 = clfLDA.predict(X_test_1)
```

```
ytrain_predict_1
```

```
array([1, 1, 1, ..., 1, 1, 0])
```

```
ytest_predict_1
```

```
array([1, 1, 1, ..., 0, 1, 1])
```

```
ytest_predict_prob_1=clfLDA.predict_proba(X_test_1)
pd.DataFrame(ytest_predict_prob_1).head()
```

	0	1
0	0.192439	0.807561
1	0.001038	0.998962
2	0.052178	0.947822
3	0.000313	0.999687
4	0.001129	0.998871

In Above We had predicted for both y train_predict_1and y test_predict_1 and also interms of Probability also

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. (8 marks)

MODEL ACCURACY SCORE-

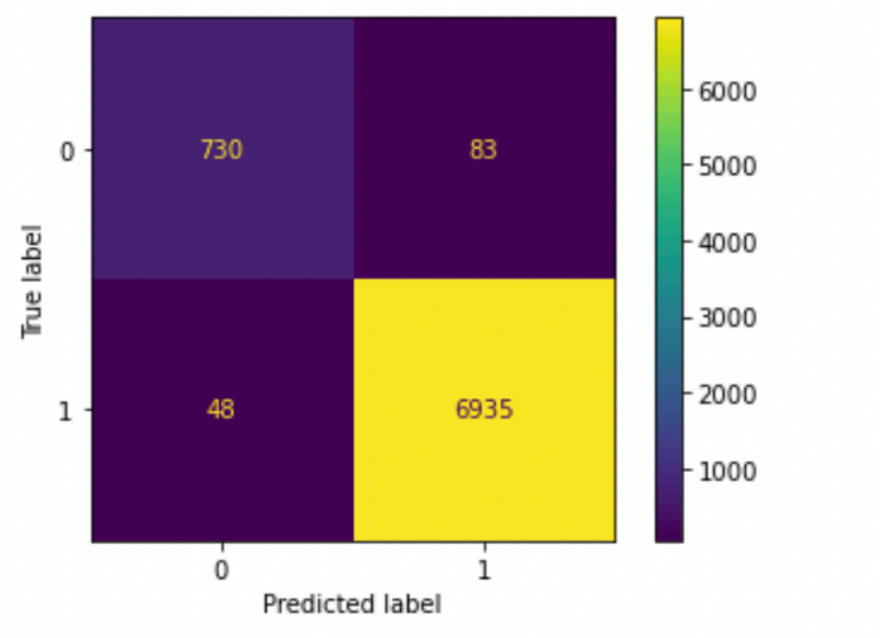
MODEL SCORE	***Logistics Regression***	***LDA***
TRAINING SET	0.98319	0,96139
TESTING SET	0,97845	0,956912

In Model Accuracy Score,The Logistic Regression has fared better than LDA in both Training and Testing.

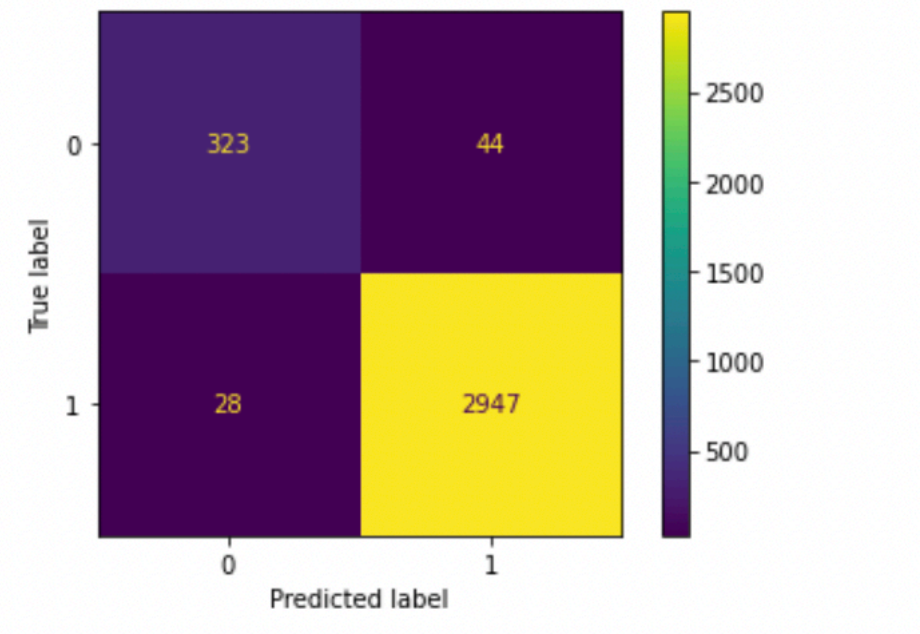
CONFUSION MATRIX

LOGISTICS REGRESSION

TRAINING MODEL



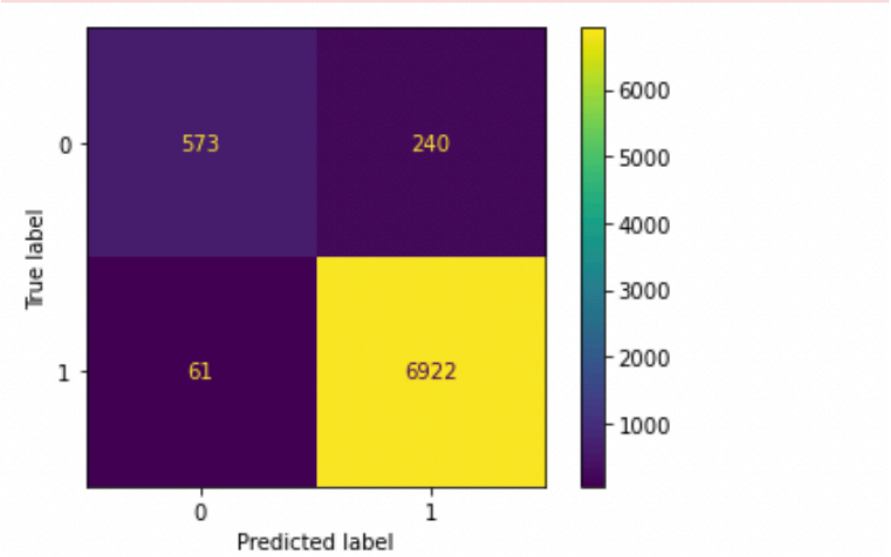
TESTING MODEL



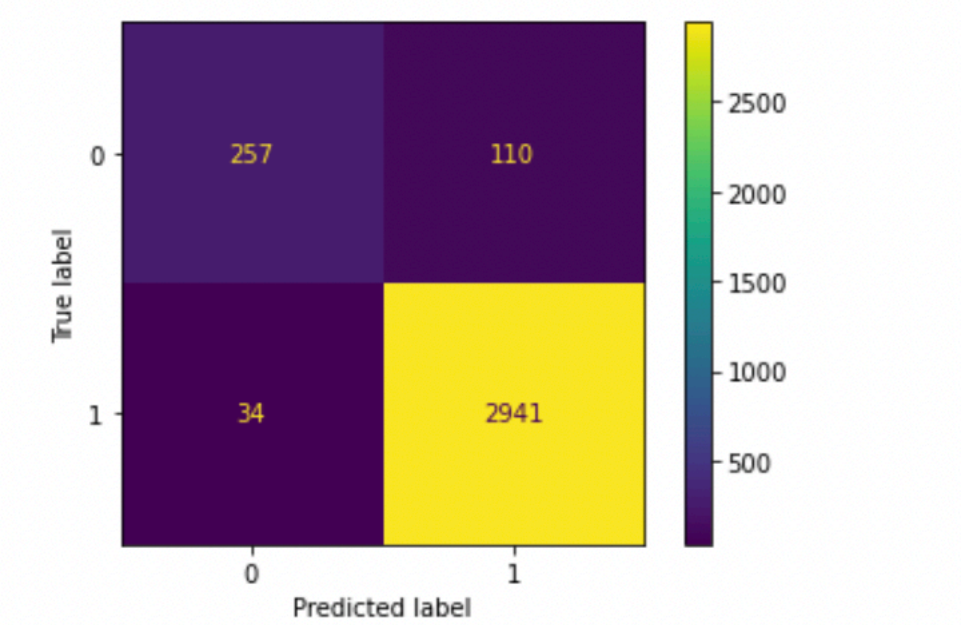
Out of total 3342 in testing phase, the model has correctly predicted 3270 with the Accuracy rate of 0.9784(~98%) which is better in terms of Analysis. The Model too predict the variable in Higher Accuracy.
The Model deals with Survival not Diagnostics so here the F- score plays an Major Role

Linear Discriminant Analysis

Training Model



Testing Model



Out of total 3342 in testing phase, the model has correctly predicted 3198 with the Accuracy rate of 0.9569(~95%) which is better in terms of Analysis
The Model deals with Survival not Diagnostics so here the F- score plays an Major Role.

Classification report

LOGISTICS REGRESSION

	precision	recall	f1-score	support
0	0.94	0.90	0.92	813
1	0.99	0.99	0.99	6983
accuracy			0.98	7796
macro avg	0.96	0.95	0.95	7796
weighted avg	0.98	0.98	0.98	7796

0	0.92	0.88	0.90	367
1	0.99	0.99	0.99	2975
accuracy			0.98	3342
macro avg	0.95	0.94	0.94	3342
weighted avg	0.98	0.98	0.98	3342

The F1 score of the Model in Logistics Regression is 0.99 which is fared better in Logistic Regression and also we can check with LDA for getting Most Optimised Model

Linear Discriminant Analysis

Predicted label

```
print(classification_report(y_train_1, ytrain_predict_1))
```

	precision	recall	f1-score	support
0	0.90	0.70	0.79	813
1	0.97	0.99	0.98	6983
accuracy			0.96	7796
macro avg	0.94	0.85	0.89	7796
weighted avg	0.96	0.96	0.96	7796

```
print(classification_report(y_test_1, ytest_predict_1))
```

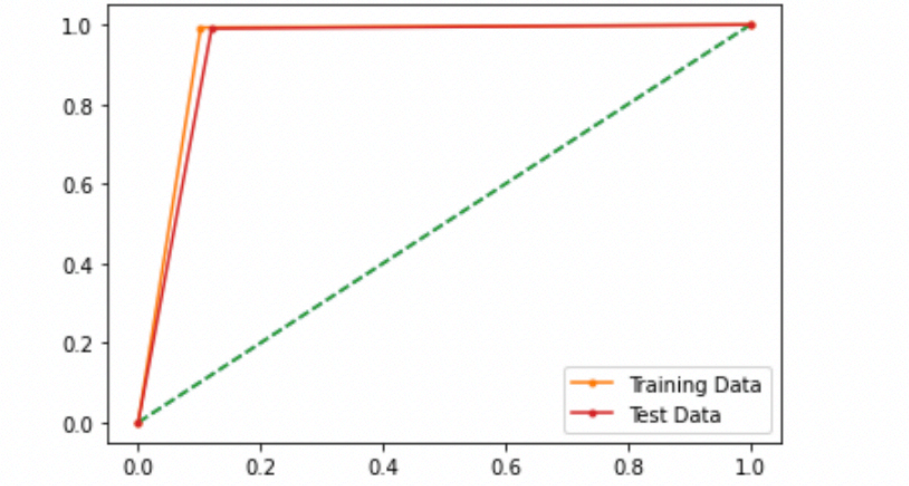
	precision	recall	f1-score	support
0	0.88	0.70	0.78	367
1	0.96	0.99	0.98	2975
accuracy			0.96	3342
macro avg	0.92	0.84	0.88	3342
weighted avg	0.96	0.96	0.95	3342

The F1 score of the Model in LDA is 0.98 which is fared better in Logistic Regression than LDA Model

LOGISTICS REGRESSION

Plot ROC curve and get ROC_AUC score

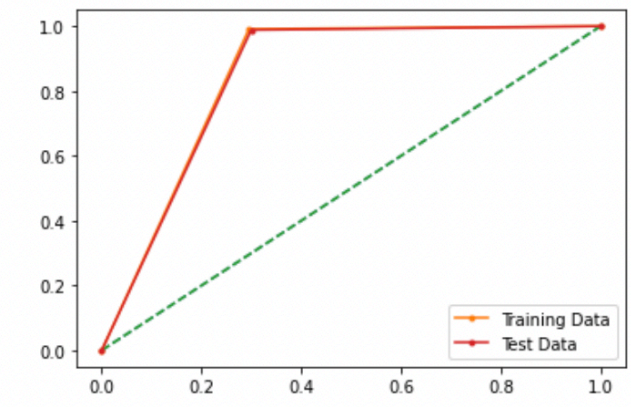
```
AUC for the Training Data: 0.946
AUC for the Test Data: 0.935
```



Linear Discriminant Analysis

Plot ROC curve and get ROC_AUC score

```
AUC for the Training Data: 0.848
AUC for the Test Data: 0.844
```



On Comparing both the Model, Logistics Regression has Fared better compared to the LDA Model and also has better AUC score

Henceforth, On Comparing both the model Logistics Regression has fared better than LDA in all the Parameters. One of the Reason is Might be the Reason of Optimisation. Hence after checking the Logistics Regression without optimisation it still had fared better than LDA model. So we can Conclude that the Logistics is the Best and Most optimised Model

Hence the Final output as follow

	dvcat	weight	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat	occRole	deploy	injSeverity	ytest_predict
2047	0.277736	-0.289793	1.303636	1.524722	-1.344494	-1.081415	-0.409529	-1.039985	-1.620935	1.169325	-0.521994	-0.799014	0.851945	1
6536	-0.764523	-0.277934	-0.767085	-0.655857	-1.344494	0.924714	0.416051	0.849170	0.676660	-1.125238	-0.521994	1.251543	0.851945	1
7905	1.319996	-0.288394	1.303636	1.524722	0.743774	0.924714	1.186593	0.849170	-0.560506	1.169325	-0.521994	-0.799014	1.577401	0
6753	0.277736	0.043308	1.303636	-0.655857	0.743774	0.924714	-1.069993	0.849170	-0.913982	1.169325	-0.521994	-0.799014	0.126489	1
5770	-0.764523	0.235655	-0.767085	-0.655857	-1.344494	-1.081415	-0.519606	-0.095408	1.206874	0.022043	-0.521994	-0.799014	-0.598967	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8264	-0.764523	-0.292044	-0.767085	1.524722	-1.344494	-1.081415	0.636206	0.849170	-0.383768	0.022043	1.915730	-0.799014	0.126489	1
0518	-0.764523	0.235655	-0.767085	-0.655857	0.743774	0.924714	0.030781	0.849170	0.146446	-1.125238	-0.521994	1.251543	-1.324422	1
487	2.362256	-0.242943	1.303636	1.524722	0.743774	0.924714	0.030781	-1.984563	-0.560506	1.169325	-0.521994	-0.799014	1.577401	0
9737	-0.764523	-0.130379	1.303636	-0.655857	-1.344494	-1.081415	-1.069993	0.849170	-0.207030	1.169325	-0.521994	-0.799014	-0.598967	1
0698	1.319996	-0.275596	1.303636	1.524722	0.743774	-1.081415	0.416051	0.849170	-1.444197	1.169325	1.915730	-0.799014	0.851945	1

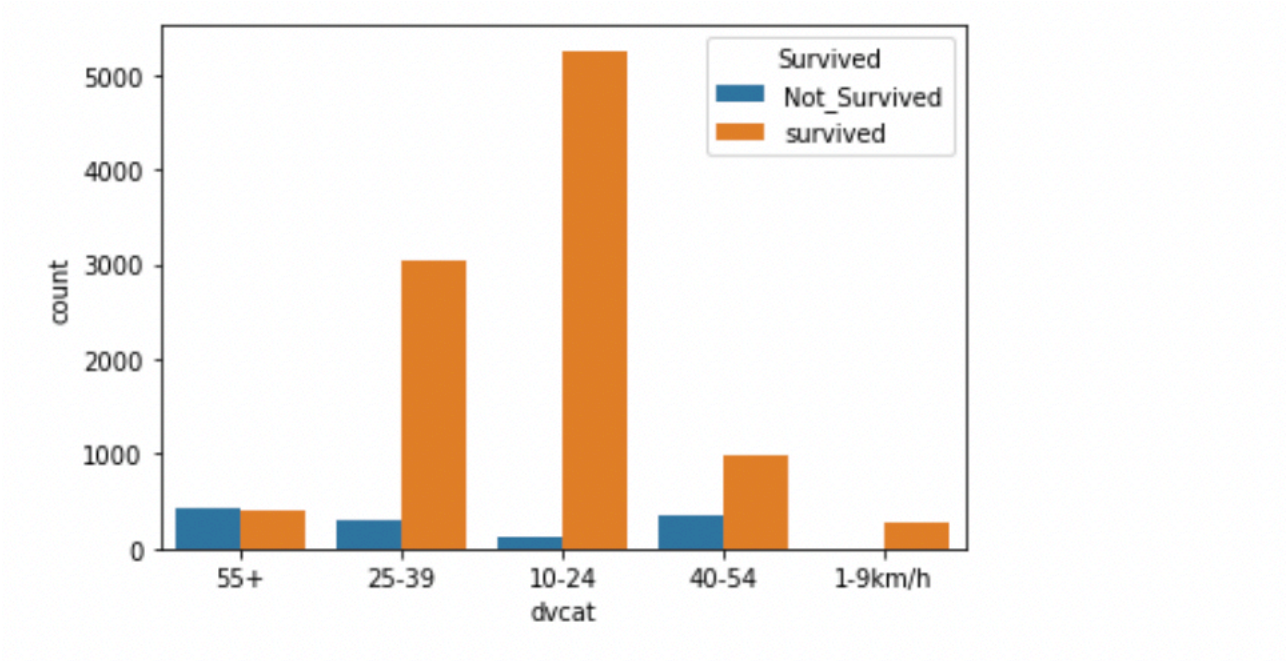
342 rows × 14 columns

2.4) Inference: Based on these predictions, what are the insights and recommendations? (6 marks)

Based on the Model,

Insights and Recommendation

- The First is the Speed Category, when ever the speed is low, The survival rate is higher henceforth the speed plays the role in Survival rate .The higher speed the higher is the chance of Death. So recommendation is there should be definitely the speed limit. Below 55 can



definitely reduce the death rate

- In terms of Weight Category,Lower the weight survival rate is also low in the above data. So recommendation here is to increase the Weight can reduce the Death rate and Increase the Survival rate
- In terms of Airbag, deployment and seat belt.The Survival has become higher in terms of Vehicle containing the Airbag,Deployment and Seat belt.So the Government is to advice the car maker to provide the Air bag and Seat Belt as compulsory in their Vehicle and it will increase the survival rate
- Interms of Ink severity also whenever the People have essential such as seat belt, Airbag the Injseverity is very low which in turn has produced higher survival rate. The Government can open some First aid centre in Main Highways so that they can Increase the Survival rate of Injured person
- Interms of Year passed, the Vehicle survival rate got Increased which mean that each vehicle should have certain regulation before launching the Vehicle which will definitely increase the Survival rate