Capstone Interim Report Submission

Batch details	**PGPDSE - Online June22**
Team members	**1. Anoosh Kumar** **2. Nidhi Jaiswal** **3. Pranjal Jalota** **4. Rushika Bokde** **5. Sylesh JL**
Domain of Project	**Insurance**
Proposed project title	**Car Insurance Claim Prediction**
Group Number	**Group-03**
Team Leader	**Nidhi Jaiswal**
Mentor Name	**Ms. Vibha Santhanam**

Date: 12th March 2023

Signature of the Mentor **Signature of the Team Leader**

Table of Contents

Business Objective / Understanding

Business Problem Goals (Problem Statement)

Business Problem Understanding :

An insurance policy is an agreement between an insurance provider company and policyholder, wherein the company is liable in providing the guaranteed compensation for a specific loss or damage when a certain amount of premium is paid by the policyholder for taking the insurance with that company.

For an instance, if someone pays a premium of Rs. 6000/- every year for car insurance premium with a cover of Rs. 190,000/-. Unfortunately, in the event of an unexpected accident and the car is damaged, In such a case the insurance company will bear the cost for the damage happened up to Rs. 190,000.

Considering that the company charges a premium of only Rs. 3000/- per annum, the concept of probability plays an important role here. In other words, there might be thousands of customers paying a premium of Rs. 6000 every year just to cover the cost of the policy.

Business Objective :

Our objective is to help the Insurance company to understand the behavior of customers and predict a future trait if a car policy holder will claim his insurance or not based on the insurance data provided.. With the above objective we aim to develop a prediction model that helps the insurance company to understand the car policyholders behavior and classify if the policy holders are likely to claim insurance.

About Dataset : The Dataset contains information on policyholders having the attributes like policy tenure, age of the car, age of the car owner, the population density of the city, make and model of the car, power, engine type, etc, and the target variable indicating whether the policyholder files a claim in the next 6 months or not.

Industry Review

Current practices :

Due to the rapid modernization, a surge in the usage of number of cars is seen in the last two decades compelling the insurance companies in utilizing most advanced data techniques in order to predict whether a policyholder opts for the claim or not. In this quest companies started using the machine learning algorithms in their business, Major categories in where the companies currently using ML models include behaviour of driver monitoring system and in depth market analysis on Data of insurance claims based on consumer previous behaviour, Example of insurance analytics is the model that identifies the severity of claim and total amount of funds to be released for various scenarios of accident using Linear regression or multiple linear regression models.

Model focus on examining the machine learning methods that are the most suitable method for claim prediction with big training data and many missing values.Some of the previous recent studies that applied some machine learning models in the insurance industry show that the XGBoost model is the best model for classification in the insurance industry. They used a database comprised of 2767 and 30,240 observations, respectively, while shows that the naïve Bayes is an effective model for claims occurrence prediction. Here we use an ML classification model that contains around fifty eight thousand (58592) observations with 44 variables. The results confirm the scalability of the random forest. Hence, the random forest model can be used to solve big data problems related to data volume.

Background Research :

To understand the problem, it is essential to understand the insurance claims forecast, big data, ML, and classification. We explore the following terms.

The vast amount of data to determine the probability of claims occurrence makes a claim prediction issue require big data models. Thus, there is a need for an effective approach and a more reliable ML model to assess the danger that the driver poses to the insurance provider and the probability of filing a claim in the coming year, a model that can read and interpret vast databases containing thousands of consumer details provided by the Porto Seguro insurance company.

Porto Seguro is one of the biggest car and homeowner insurance firms in Brazil.

Porto Seguro claims that their automotive division’s mission is to customize insurance quotes based on the driver's ability. They believe that effective techniques can be applied for more accurate results to predict claims occurrence in the coming year. Thus, they provided the dataset containing 59 variables with 1,488,028 observations4. These observations include customer information that the company collected over several years.

Literature Survey :

Oyugi, studied secondary data of claim amounts obtained from certain Insurance company in Nairobi, regarding their motor comprehensive policy.These researchers fitted Exponential, Gamma, Weibull and lognormal probability distributions and concluded that lognormal distribution is suitable for modeling the data under study.

Burney and Hashmi have discussed different claim amount distributions as well as selection methods of distribution functions for claim amounts, seeBurney at al.". Talangtam, et al. studied in order to model the data set of claim amounts of motor insurance using finite mixture lognormal distributions, and estimating parameters by EM algorithm. To decide best fitted model Kolmogorov Smirnoff (K-S) and A-D tests were used.

Meyres , studied the data of 250 claims to decide suitable statistical probability distribution which could be used for modeling the data of claim amounts. The researcher fitted Gamma, Weibull and lognormal probability distributions to the data of claim amounts under study. The parameters of the fitted distribution were estimated by the method of maximum
likelihood.

References :

https://dalpozz.github.io/static/pdf/Claim_prediction.pdf

https://arxiv.org/pdf/2204.06109.pdf

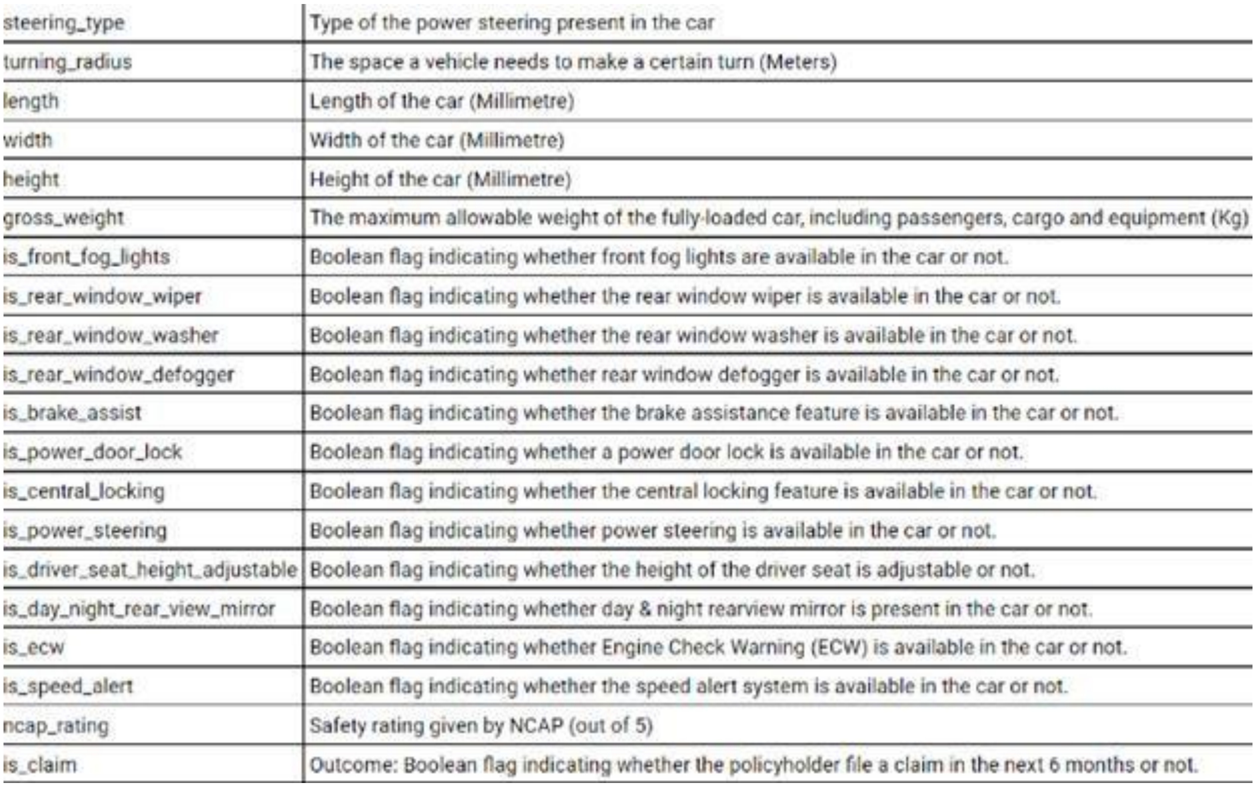
https://www.jatit.org/volumes/Vol98No22/8Vol98No22.pdf

https://www.virtusa.com/perspectives/article/predictive-analytics-in-insurance-claims

https://www.genpact.com/insight/analytics-in-claims-how-to-gain-better-faster-consumable-insights-using-augmented-intelligence

Dataset and Domain :

Data Dictionary :



steering_type	Type of the power steering present in the car
turning_radius	The space a vehicle needs to make a certain turn (Meters)
length	Length of the car (Millimetre)
width	Width of the car (Millimetre)
height	Height of the car (Millimetre)
gross_weight	The maximum allowable weight of the fully-loaded car, including passengers, cargo and equipment (Kg)
is_front_fog_lights	Boolean flag indicating whether front fog lights are available in the car or not.
is_rear_window_wiper	Boolean flag indicating whether the rear window wiper is available in the car or not.
is_rear_window_washer	Boolean flag indicating whether the rear window washer is available in the car or not.
is_rear_window_defogger	Boolean flag indicating whether rear window defogger is available in the car or not.
is_brake_assist	Boolean flag indicating whether the brake assistance feature is available in the car or not.
is_power_door_lock	Boolean flag indicating whether a power door lock is available in the car or not.
is_central_locking	Boolean flag indicating whether the central locking feature is available in the car or not.
is_power_steering	Boolean flag indicating whether power steering is available in the car or not.
is_driver_seat_height_adjustable	Boolean flag indicating whether the height of the driver seat is adjustable or not.
is_day_night_rear_view_mirror	Boolean flag indicating whether day & night rearview mirror is present in the car or not.
is_ecw	Boolean flag indicating whether Engine Check Warning (ECW) is available in the car or not.
is_speed_alert	Boolean flag indicating whether the speed alert system is available in the car or not.
ncap_rating	Safety rating given by NCAP (out of 5)
is_claim	Outcome: Boolean flag indicating whether the policyholder file a claim in the next 6 months or not.

Variable categorization :

We have 28 categorical and 16 numerical columns in our dataset.

Categorical columns :

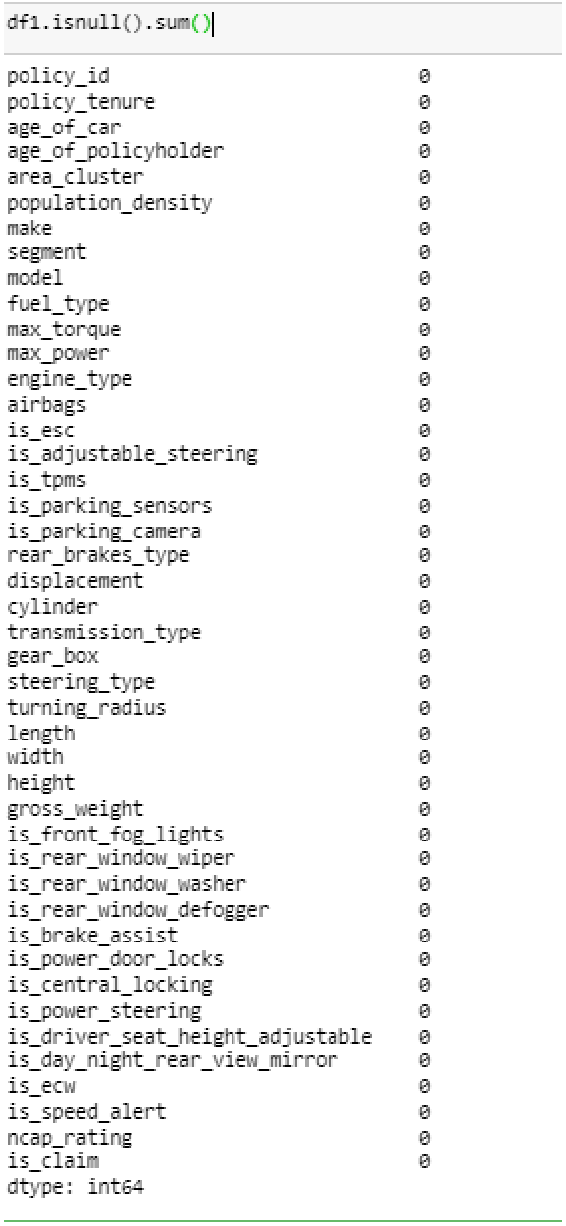
['policy_id', 'area_cluster', 'segment', 'model', 'fuel_type', 'max_torque', 'max_power', 'engine_type', 'is_esc', 'is_adjustable_steering', 'is_tpms', 'is_parking_sensors', 'is_parking_camera', 'rear_brakes_type', 'transmission_type', 'steering_type', 'is_front_fog_lights', 'is_rear_window_wiper', 'is_rear_window_washer', 'is_rear_window_defogger', 'is_brake_assist', 'is_power_door_locks', 'is_central_locking', 'is_power_steering','is_driver_seat_height_adjustable', 'is_day_night_rear_view_mirror', 'is_ecw', 'is_speed_alert']

Numerical columns :

['policy_tenure', 'age_of_car', 'age_of_policyholder','population_density', 'make', 'airbags', 'displacement', 'cylinder','gear_box', 'turning_radius', 'length', 'width', 'height', 'gross_weight', 'ncap_rating', 'is_claim']

Pre Processing Data Analysis :

We do not have any null values in our dataset.



```
df1.isnull().sum()
```

```
policy_id                           0
policy_tenure                       0
age_of_car                          0
age_of_policyholder                 0
area_cluster                        0
population_density                  0
make                                0
segment                             0
model                               0
fuel_type                           0
max_torque                          0
max_power                           0
engine_type                         0
airbags                             0
is_esc                              0
is_adjustable_steering              0
is_tpms                             0
is_parking_sensors                  0
is_parking_camera                   0
rear_brakes_type                    0
displacement                        0
cylinder                            0
transmission_type                   0
gear_box                            0
steering_type                       0
turning_radius                      0
length                              0
width                               0
height                              0
gross_weight                        0
is_front_fog_lights                 0
is_rear_window_wiper                0
is_rear_window_washer               0
is_rear_window_defogger             0
is_brake_assist                     0
is_power_door_locks                 0
is_central_locking                  0
is_power_steering                   0
is_driver_seat_height_adjustable    0
is_day_night_rear_view_mirror       0
is_ecw                              0
is_speed_alert                      0
ncap_rating                         0
is_claim                            0
dtype: int64
```

Alternate sources of data that can supplement the core dataset :

We have developed three new feature engineering yields by combining length , breadth and height into volume , derived max power and max torque using the scientific formulas, and found a combined score of all less insignificant columns into a single column., which is discussed in detail in the below feature engineering section.

Project Justification :

Project Statement :

The insurance company wants to understand if a policy holder will claim his/her policy in next 6 months based on the data survey to take some business decisions. The dataset provided contains the 44 variables out of which 43 are independent variables that specify the details of the car such as age of car, model, torque , power , the sensors the car has equipped with , ncap rating of the car for safety, dimensions of the car etc , along with this the details of the policyholder such as age of the policyholder is also shared. The one remaining column is Dependent variable (is_claim) which will help in fetching whether a particular policy will be claimed or not based on the various given independent variables' conditions.

Complexity involved :

The vital practice in an Insurance industry is to set the premium before the contract inception. To decide an accurate premium for the consecutive years in an insurance company, a precise and reliable estimate of the number of claims occurrences and the total claim amounts is humongously important. The claims history is the backbone for new insurance products which driven to the market expecting huge margins of attraction and profits. But not large volume of products are released because manual analysis are still performed in many insurance sector. Faster and accurate predictions are still an ordeal in this sector. We are aiming to provide a prediction model with Enhanced accurate forecast of claims which can help insurers maneuver with new release of products with multiple offerings to insurers and added benefits to customer as well.

Data obtained is highly imbalanced and its hard to make predictions for such an imbalanced data.

Project Outcome :

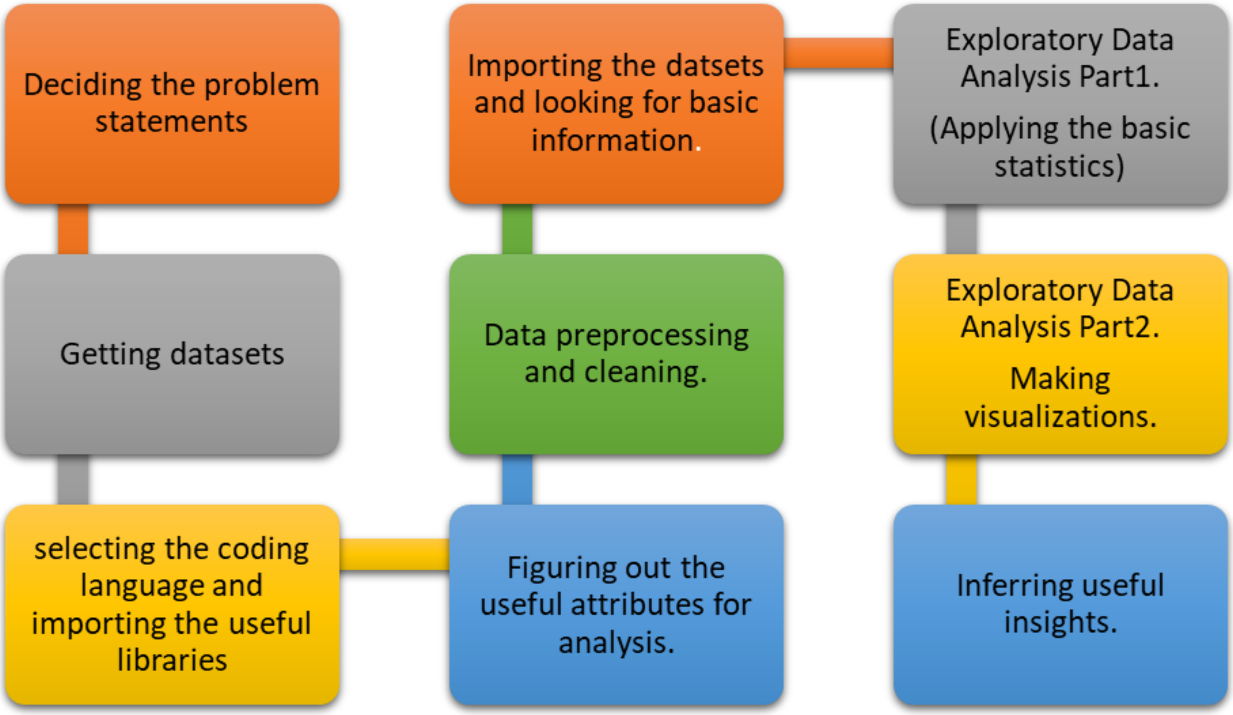
Our ultimate goal of this project is building a classification model. A classification model helps to make some inference from the input values given for training. It will predict the class labels of new data. Here we mainly focus on those independent variables that describe only the various prospects of the car which include cars's build , shape , mileage, age, safety rating, Engine power, torque it produces, availability of various sensors etc.

We are implementing complex machine learning models in predicting whether a policy holder's likelihood in claiming the insurance in next 6 months. Here we use variety of classification algorithms techniques like Logistic Regression, Naïve Bayes, K-Nearest Neighbours, Decision Tree or Random Forest to classify a particular policy ID is a "claim" or "not a claim". The idea is to build a machine learning solution which is diversified, hence this project will have more diversity in combining models and various ensemble techniques for an accurate claim prediction.

Applying ML analytics in insurance is the same as in other industries—to optimize marketing strategies, improve the business, enhance the income, and reduce costs. This paper presented several ML techniques to efficiently analyze insurance claim prediction and compare their performances using various metrics. We proposed a solution using ML models to predict claim occurrence in the next year and to adjust the insurance prices fairly to the client's ability, and used relevant personal information. Thus, insurance companies can make automotive insurance more accessible to more clients through a model that creates an accurate prediction. The accuracy of the prediction of claims can have a significant effect on the real economy. It is essential to routinely and consistently train workers in this new area to adapt and use these new techniques properly. Therefore, regulators and policymakers must make fast decisions to monitor the use of data science techniques, maximizing efficiency and understanding some of these algorithms' limits.

Data Exploration (EDA)

EDA methodology workflow :


Deciding the problem statements
Getting datasets
selecting the coding language and importing the useful libraries
Figuring out the useful attributes for analysis.
Data preprocessing and cleaning.
Importing the datsets and looking for basic information.
Exploratory Data Analysis Part1.
(Applying the basic statistics)
Exploratory Data Analysis Part2.
Making visualizations.
Inferring useful insights.

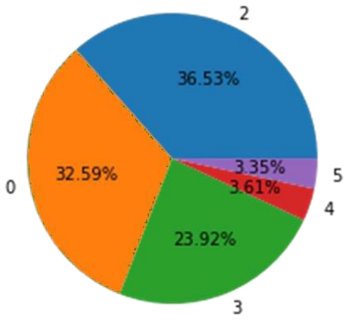
Relationship between variables :

To find the possibility of a meaningful relationship between numerical attributes, the hat map was plotted. From the heat map, we can infer that some columns like height, displacement, turning radius, length width, height, gross weight, and NCAP rating are related to each other.

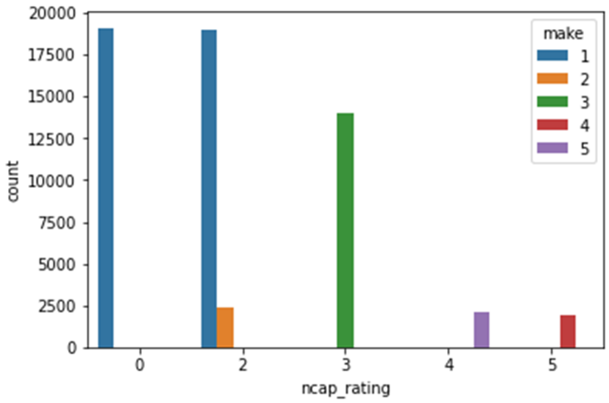
From plot (A) we can say that all the moderate or strong linear relations are positive or directly proportional to each other.

We used different plots to infer relations between different attributes of different types. From our exploratory data analysis we were able to infer some important results which are:

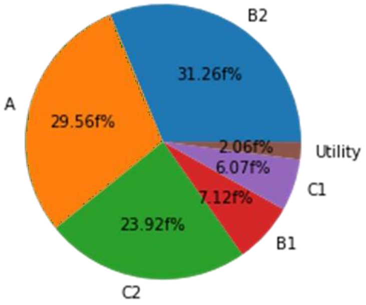
1. We can see that most of the cars have an **NCAP** rating of 2, 0, and 3 whereas very less have NCAP ratings of 4 and 5 also no cars have an NCAP rating of 1. This means most cars are not having good safety ratings.



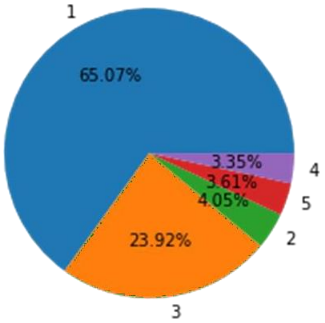
2. The rating is very much dependent on the manufacturer. Manufacturer 1 has the lowest ratings, followed by 2, then 3,5, and 4 respectively. Hence the cars from **Manufacturer** 4 have the highest **NCAP rating**.



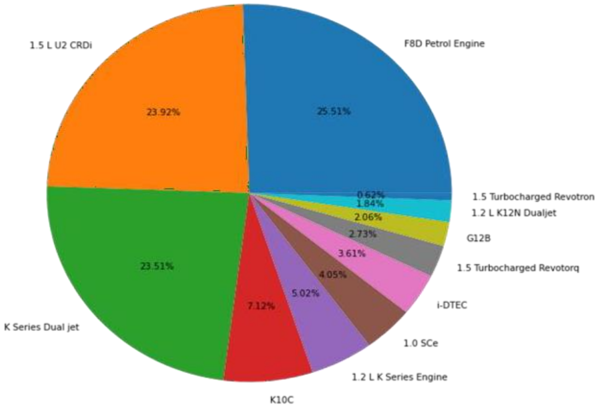
3. We can see that most cars are from A, B2 and C2 **Segment** whereas very few are from B1,C1 and utility. Hence we can say most of the population goes for a specific segment of cars. This may be due to various factors for example demographics, utility, economic distribution, etc.



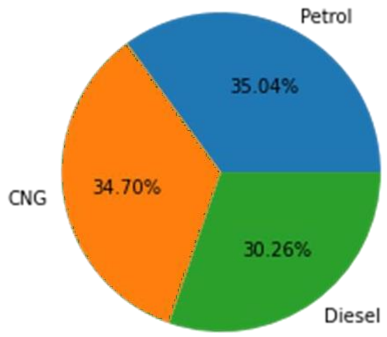
4·	**Manufacturer** 1 has the maximum number of cars whereas very less cars are from companies 2, 5, and 4. This may be because manufacturer 1 is more popular. The popularity can be based on less cost, better service, low maintenance, better features, etc.



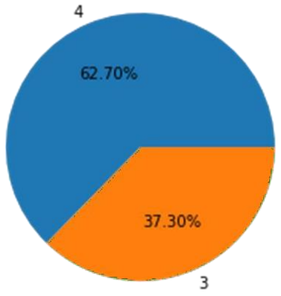
5·	F8D Petrol engine, 1.5 L U2 CRDi, and K Series Dual jet are the most commonly used **Engine types**.



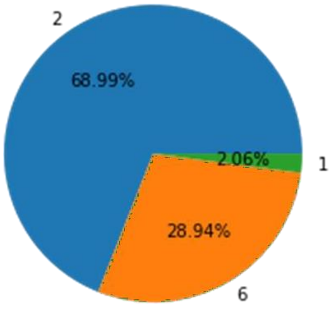
6· Cars use nearly all types of **fuel types** but petrol is the maximum among them. Since petrol is the oldest type of fuel among the given, this was expected.



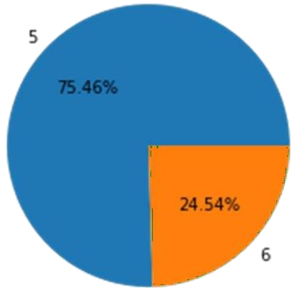
7· Most cars have 4 **Cylinders** present in the engine of the car. This can be due to the types of engines mostly used.



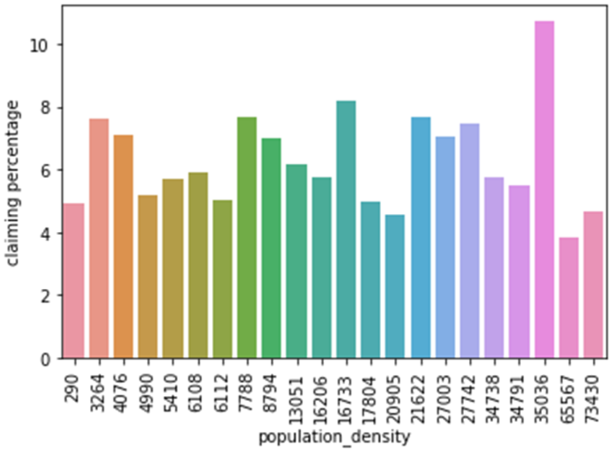
8· Most of the cars have 2 **Airbags** installed in the car. This means that people are generally going for lower or middle models for the same segment car because it costs less. More airbags are present only in higher or more expensive models of cars. This might be another reason explaining why the NCAP rating is mostly 0, 2 and 3

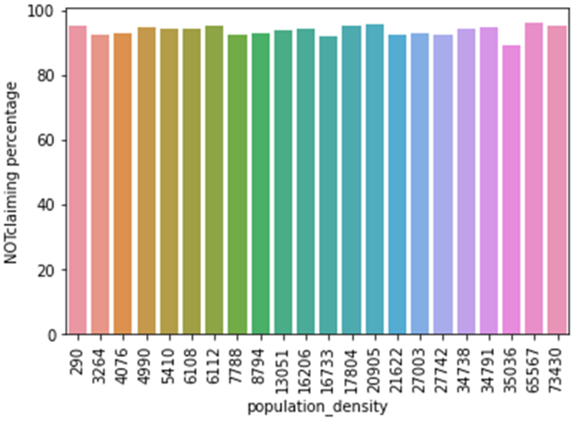


9· Most of the cars have 5 **Gears** installed in the car. This may be because more gears are present only for cars of higher segments which are generally expensive.



10. From below graphs we can say that **Population density** do not have any significant effect on claiming the insurance. It is mostly uniform throughout.



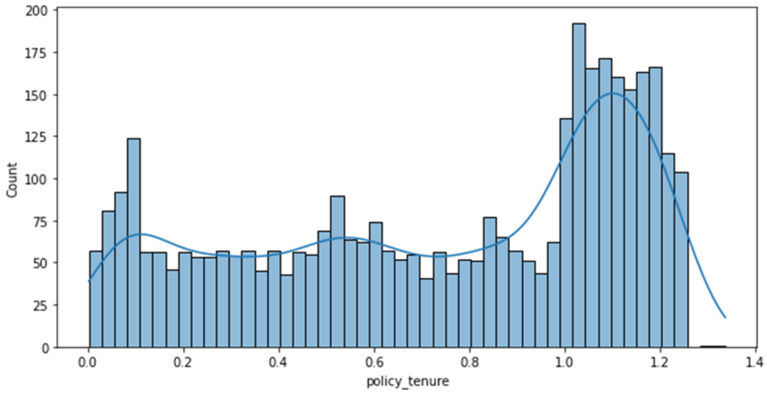


11· From the below plot we may say that there is a steep increase in claiming for the policies whose tenure is more than 1.

Now there can be several reasons for that, like, since the **Tenure** is longer the parameter supporting claim have more time to get fulfilled (older the car is, more is the duration for something getting damage), also another factor can be false claiming due to increase in number of premiums to be paid, etc.

The reason behind this can be found by proper field study.

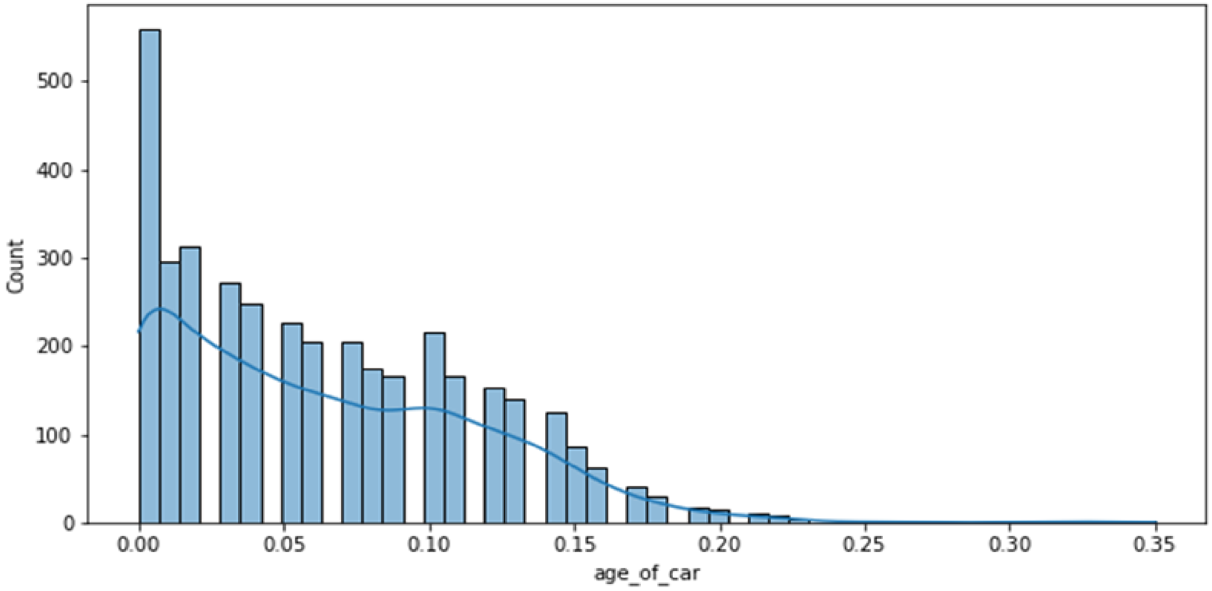
For now ==> We can conclude that in general claiming is more for policies which have normalized tenure more than 1.



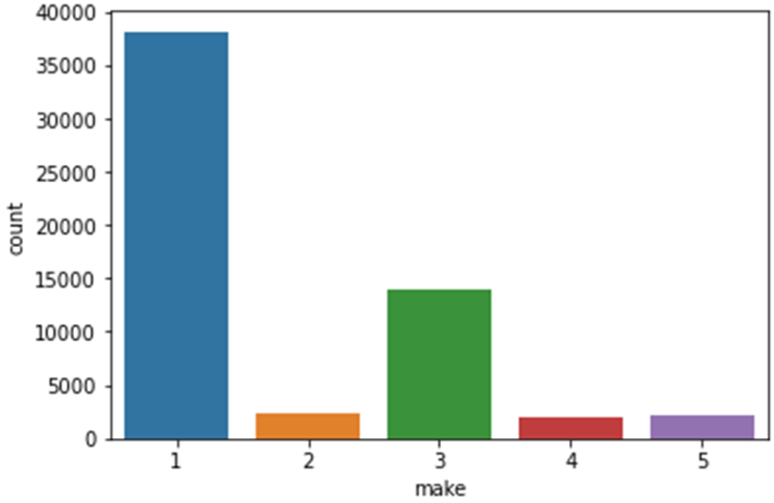
12· From the below plot we can say that among the people going for the claim, the trend is that as the **Age of the car** increases the number of claims decreases.

Now this may be just because the number of cars with less age are much more than those with more age.

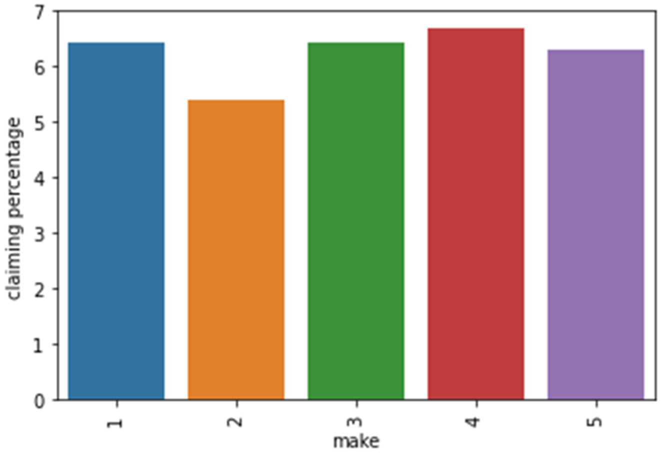
Due to above we can go for a different method, lets look at the claiming percentage at every interval 0.5 for age.



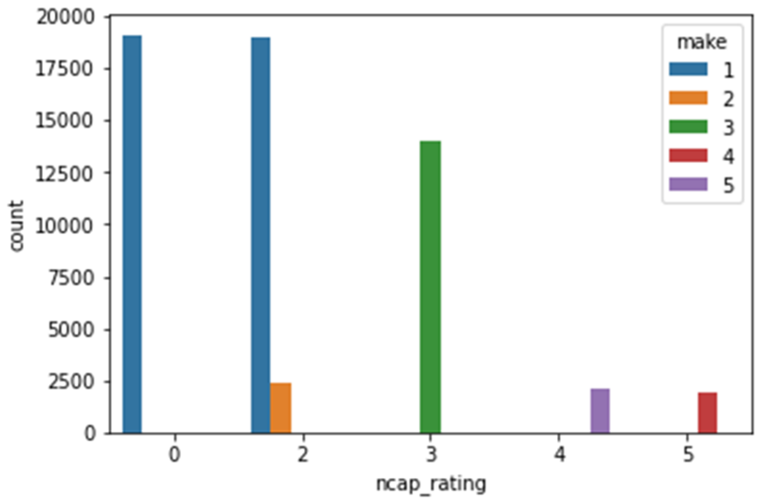
13. Here we can see that the **Manufacturer** 1 has most number of cars or customers in the given dataset followed by the manufacturer 3.



14. From below we may say that the claiming percentage for **Manufacturer** 2 is least when compared to others. This can be due to better quality of car or maybe some other reason.



15. From here we can say that the rating is very much dependent on the manufacturer. The manufacturer 1 has lowest ratings, followed by 2, then 3,5, and 4 respectively. Hence the cars from manufacturer 4 has highest **ncap rating**.

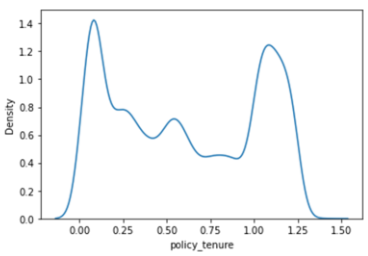


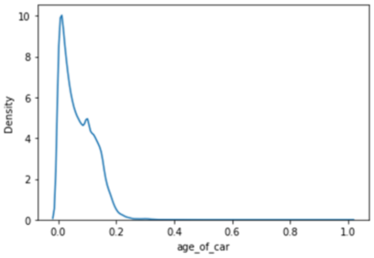
Hence from here, we can see that multicollinearity exists for attributes: make, airbags, displacement, cylinder, gearbox, turning radius, length, width, height, gross weight, and NCAP rating.

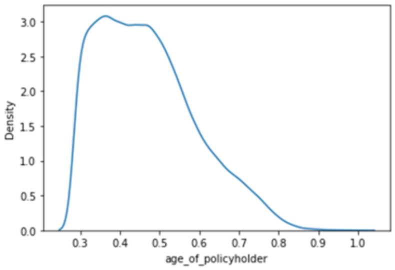
Hence, we will be trying to reduce the multicollinearity by removing one or more of the correlated predictors, Combining correlated predictors, Regularization, Centering, etc. The method to be used depends on the specifics of the data and the goal of the analysis, some methods may be more appropriate than others.

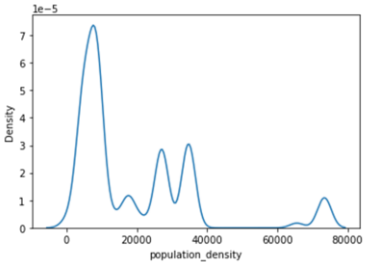
Distribution of variables :

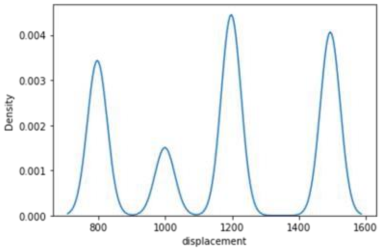
From our distribution plots, we are able to say that the attributes are not normally distributed and are highly skewed. Hence we will be needing to transform that distribution by using transformation techniques like a central limit theorem, box cox, etc. We may also go for non-parametric tests as it does not need data to follow a specific distribution.

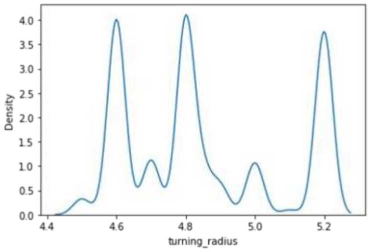


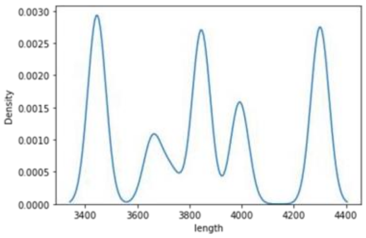


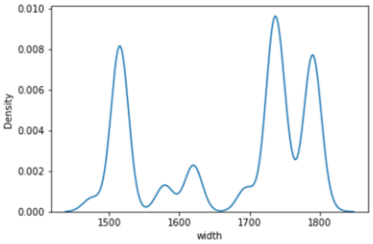


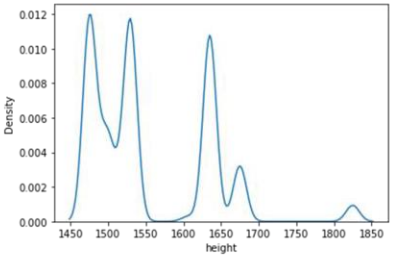


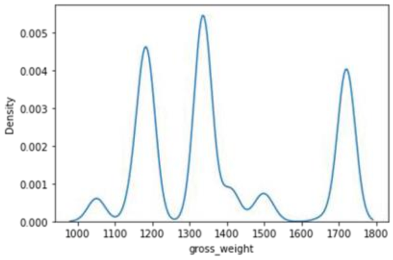


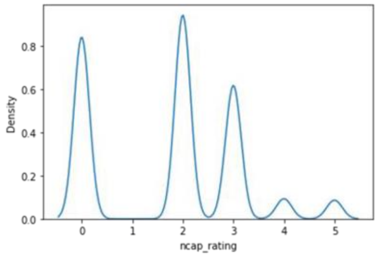




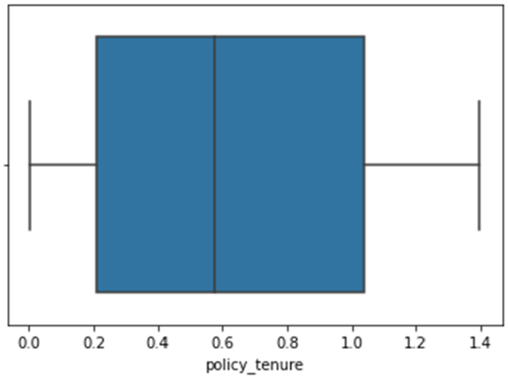


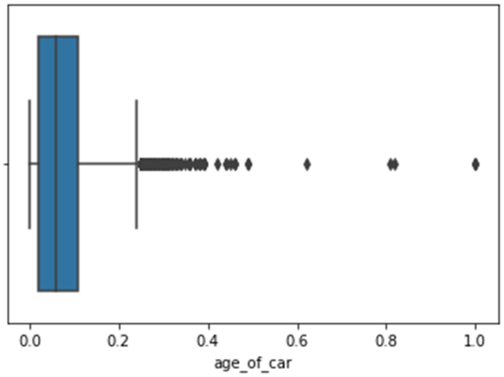


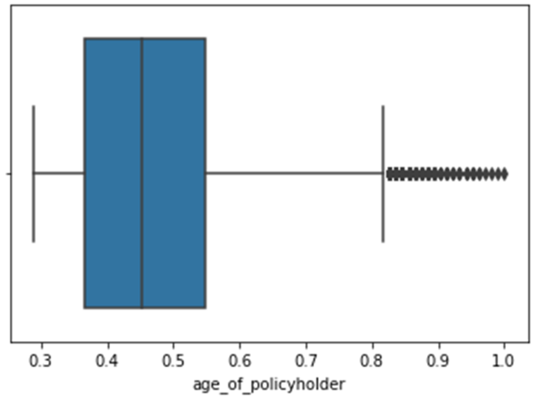


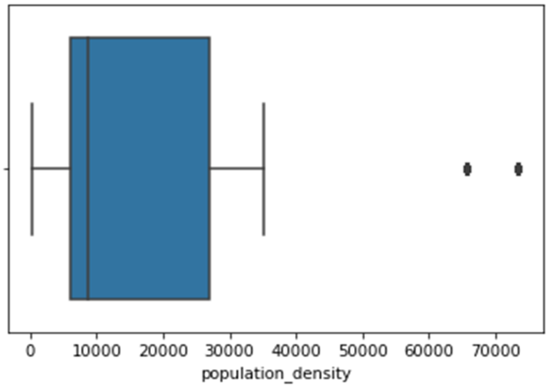


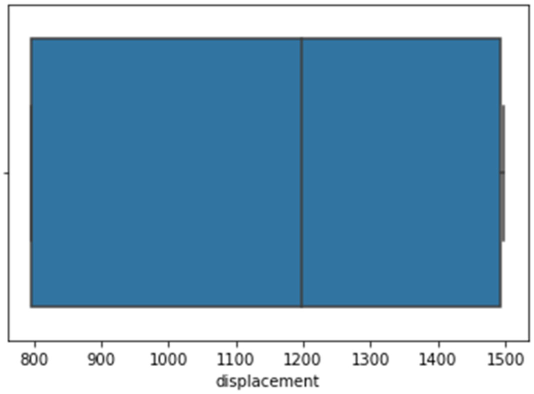
Presence of outliers and its treatment :

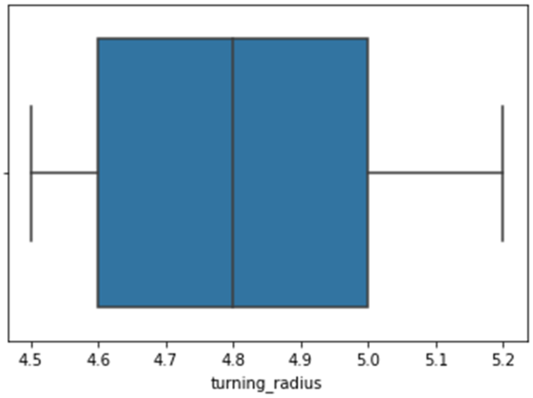


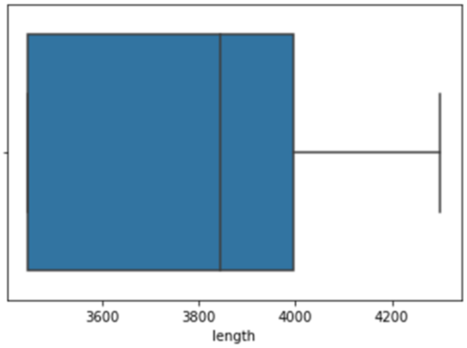


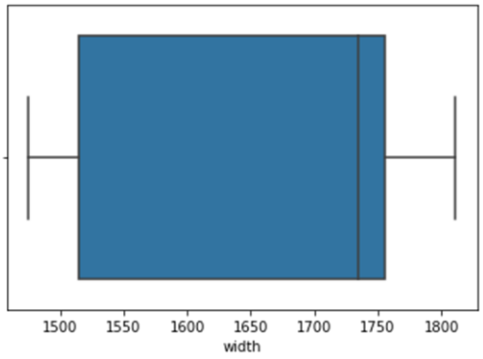

0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
age_of_policyholder

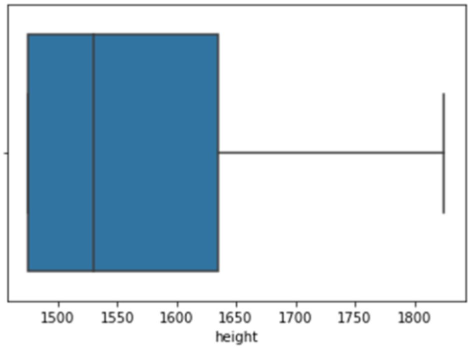

0
10000
20000
30000
40000
50000
60000
70000
population_density

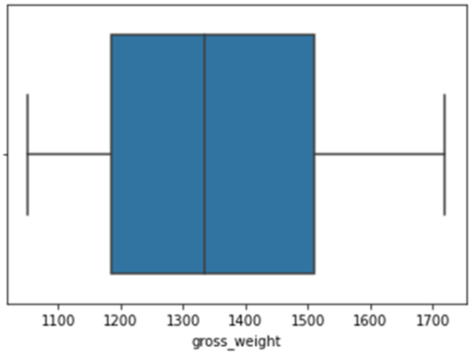

800
900
1000
1100
1200
1300
1400
1500
displacement

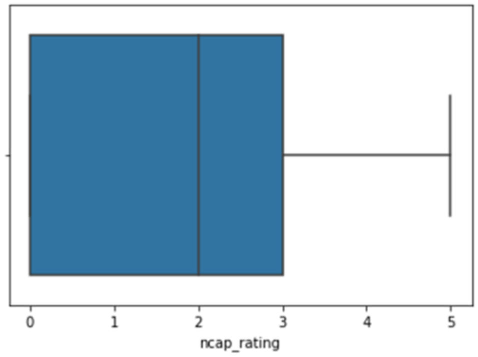

4.5
4.6
4.7
4.8
4.9
5.0
5.1
5.2
turning_radius


3600
3800
4000
4200
length


1500
1550
1600
1650
1700
1750
1800
width


1500
1550
1600
1650
1700
1750
1800
height


1100
1200
1300
1400
1500
1600
1700
gross_weight



We can see from plots that age_of_car, age_of_policyholder and population_density have outliers present in them.

Feature Engineering :

We have developed three new feature engineering techniques by combining length , breadth and height into “volume” , derived max power and max torque using the scientific formulas, and found a combined score of all less insignificant columns into a single column called “feature_score”.

Finding the volume feature :

df_featured['volume']=df_featured['length'] * df_featured['width'] * df_featured['height']

df_featured.drop(['length','width','height'],axis=1,inplace=True)

Using the volume column of the dataset one can infer that the larger the volume of car is the bigger is the car specifying its segment making it fall into SUV category, similarly the lesser the volume of car making it fall into hatchback segment.

Finding the maximum power and maximum torque of the cars :

# Calculation of max power to max torques:

# MaximumTorque = [(40.36bhp * 5252) / 6000rpm] * 1.3558179483 NewtonMeter

```
def final_max_torque(x):
    p=x.split('bhp@')[0]
    r = x.split('bhp@')[1].strip('rpm')
    p_float = float(p)
    r_float = float(r)
    return (p_float*5252*1.3558179483/6000)
```

This code defines a function called final_max_torque(x) that takes in a string x as its input. The function is performing a calculation to determine the maximum torque of an engine using the maximum power and the engine RPM. The calculation is based on the following formula:

```
# Calculation of max torque to max power :
# BHP = (60 * 3500)/(5252 * 1.3558179483) bhp = 29.49 bhp
def final_max_power(x):
    t=x.split('Nm@')[0]
    r = x.split('Nm@')[1].strip('rpm')
    t_float = float(t)
    r_float = float(r)
    return ((t_float*r_float)/(1.3558179483*5252))
```

This code defines a function called final_max_power(x) that takes in a string x as its input. The function is performing a calculation to determine the maximum power of an engine using the maximum torque and the engine RPM. The calculation is based on the following formula: BHP = (60 * 3500)/(5252 * 1.3558179483)

```
df_featured['final_max_power']=df_featured.max_torque.apply(final_max_power)
df_featured['final_max_torque']=df_featured.max_power.apply(final_max_torque)
```

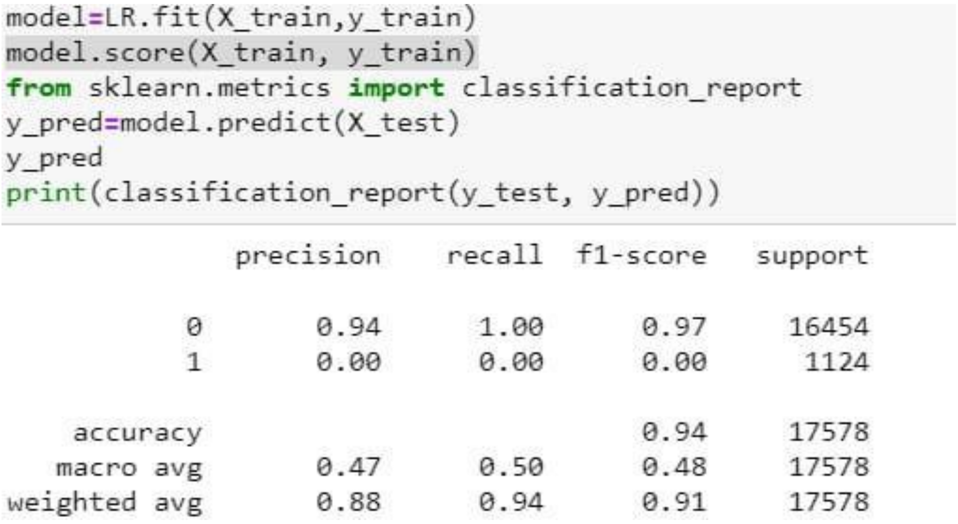
Assumptions :

1. The logistic regression assumes that there is minimal or no multicollinearity between the independent variables.
2. The logistic regression assumes that the independent variables are linearly related to the log of odds values.
3. The logistic regression requires a large sample size to predict properly.
4. Lack of strongly influencing outliers in the data
5. The Logistic regression assumes the observations to be independent of each other.

Base Model :

we have here build a model using logistic regression using scikit learn library.

Case 1



```
model=LR.fit(X_train,y_train)
model.score(X_train, y_train)
from sklearn.metrics import classification_report
y_pred=model.predict(X_test)
y_pred
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     16454
           1       0.00      0.00      0.00      1124

    accuracy                           0.94     17578
   macro avg       0.47      0.50      0.48     17578
weighted avg       0.88      0.94      0.91     17578
```

Inference :

For the above model we had imbalanced data in the target “claim” column, 93% of the target variable is non claim whereas only 7% of the target variable is claim data, due to this we can see in the model summary that the precision and f1-scores to be zero(0).

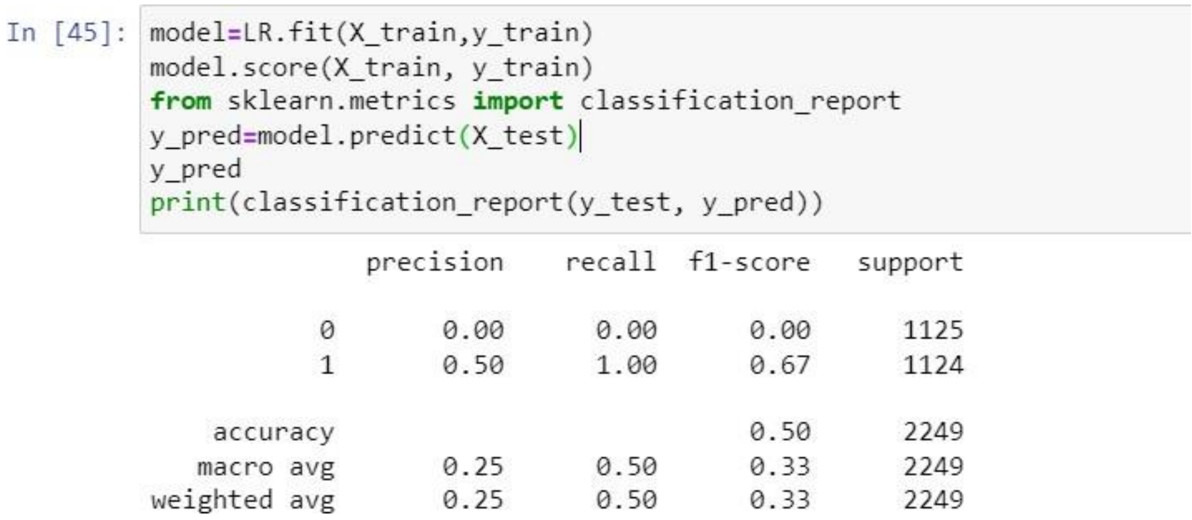
Case 2

In order to validate this we built a model (case2) with almost equal number of claim vs non claim values using manual slicing :

X_1=pd.concat((df_featured[df_featured.is_claim==0].iloc[:3748],df_featured[df_featured.is_claim==1]),axis=0).drop(['is_claim','policy_id'],axis=1)

X_1

y_1=pd.concat((df_featured[df_featured.is_claim==0].iloc[:3748],df_featured[df_featured.is_claim==1]),axis=0)['is_claim'].astype(int)



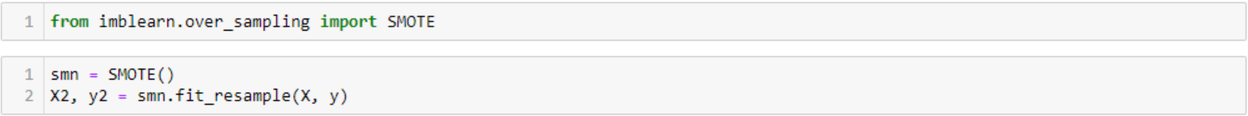
```
In [45]: model=LR.fit(X_train,y_train)
         model.score(X_train, y_train)
         from sklearn.metrics import classification_report
         y_pred=model.predict(X_test)
         y_pred
         print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1125
           1       0.50      1.00      0.67      1124

    accuracy                           0.50      2249
   macro avg       0.25      0.50      0.33      2249
weighted avg       0.25      0.50      0.33      2249
```

From the above model (Case 2 model), we can conclude that base model(Case 1 model) performance is hugely impacted by target variable imbalance.

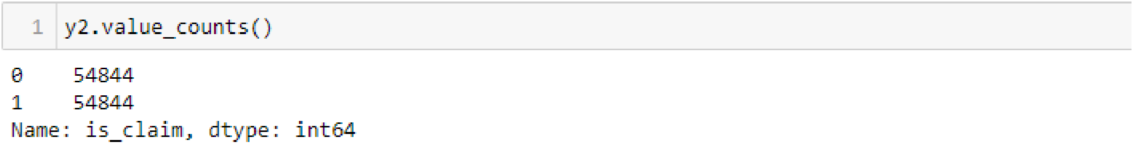
Balancing Target Variable:

We introduced SMOTE oversampling technique to increase the data dimension such that the target variable (is_claim) will be in the ratio 1:1.



```
from imblearn.over_sampling import SMOTE
```

```
smn = SMOTE()
X2, y2 = smn.fit_resample(X, y)
```

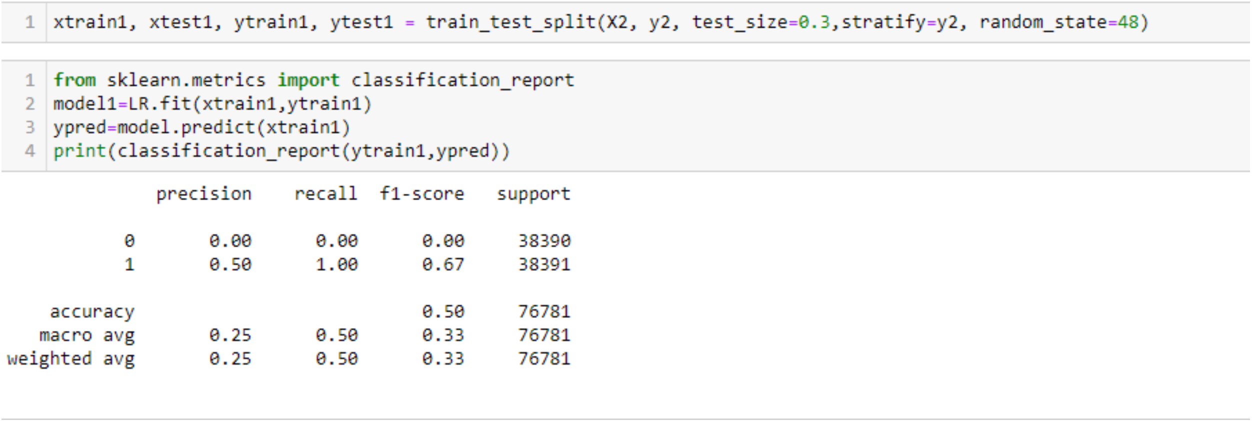


```
y2.value_counts()
```

```
0    54844
1    54844
Name: is_claim, dtype: int64
```

Model Fine Tuning:

We built the same model with new data after SMOTE and verified the results



```
xtrain1, xtest1, ytrain1, ytest1 = train_test_split(X2, y2, test_size=0.3,stratify=y2, random_state=48)
```

```
from sklearn.metrics import classification_report
model1=LR.fit(xtrain1,ytrain1)
ypred=model.predict(xtrain1)
print(classification_report(ytrain1,ypred))
```

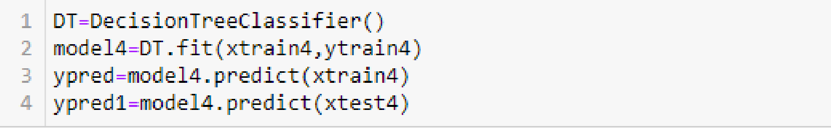
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     38390
           1       0.50      1.00      0.67     38391

    accuracy                           0.50     76781
   macro avg       0.25      0.50      0.33     76781
weighted avg       0.25      0.50      0.33     76781
```

The results were same as Case 2 base model. So balancing of target variable and better performance was achieved due to SMOTE.

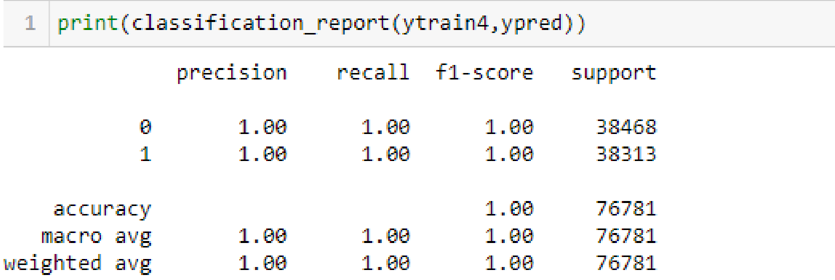
We tried the the updated atas with ensembling techniques (tree,bagging,boosting)

And model performances were obsereved.

Decision Tree:

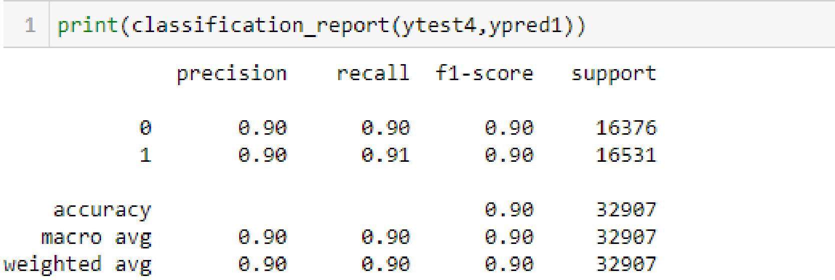


```
DT=DecisionTreeClassifier()
model4=DT.fit(xtrain4,ytrain4)
ypred=model4.predict(xtrain4)
ypred1=model4.predict(xtest4)
```



```
print(classification_report(ytrain4,ypred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38468
           1       1.00      1.00      1.00     38313

    accuracy                           1.00     76781
   macro avg       1.00      1.00      1.00     76781
weighted avg       1.00      1.00      1.00     76781
```

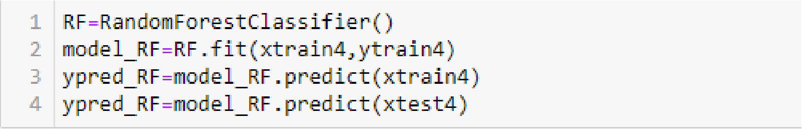


```
print(classification_report(ytest4,ypred1))
```

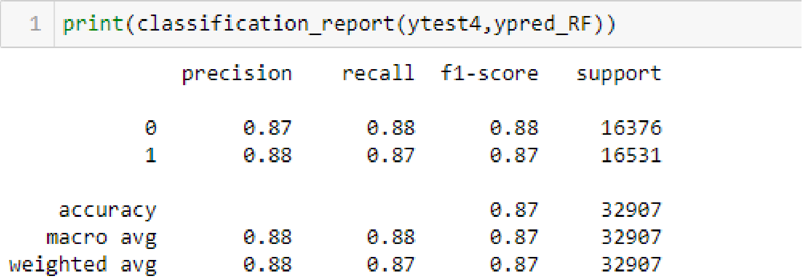
```
              precision    recall  f1-score   support

           0       0.90      0.90      0.90     16376
           1       0.90      0.91      0.90     16531

    accuracy                           0.90     32907
   macro avg       0.90      0.90      0.90     32907
weighted avg       0.90      0.90      0.90     32907
```

Performance is model was tremendously improved but the model was overfitted as the performance of train was better than test.

Random Forest:



```
RF=RandomForestClassifier()
model_RF=RF.fit(xtrain4,ytrain4)
ypred_RF=model_RF.predict(xtrain4)
ypred_RF=model_RF.predict(xtest4)
```

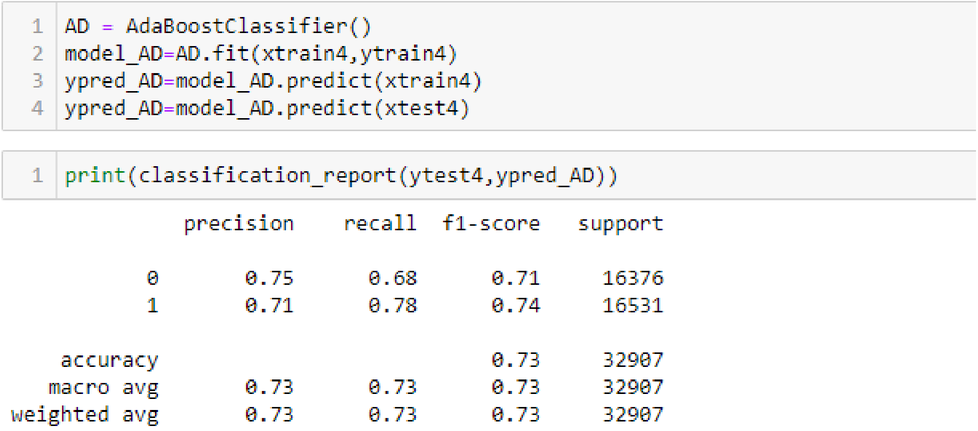


```
print(classification_report(ytest4,ypred_RF))
```

```
              precision    recall  f1-score   support

           0       0.87      0.88      0.88     16376
           1       0.88      0.87      0.87     16531

    accuracy                           0.87     32907
   macro avg       0.88      0.88      0.87     32907
weighted avg       0.88      0.87      0.87     32907
```

Its performance is comparitiverly less than Decision tree and as the accuracies were marginally less when compared to overfitted Decision tree model.

Ada Boost:



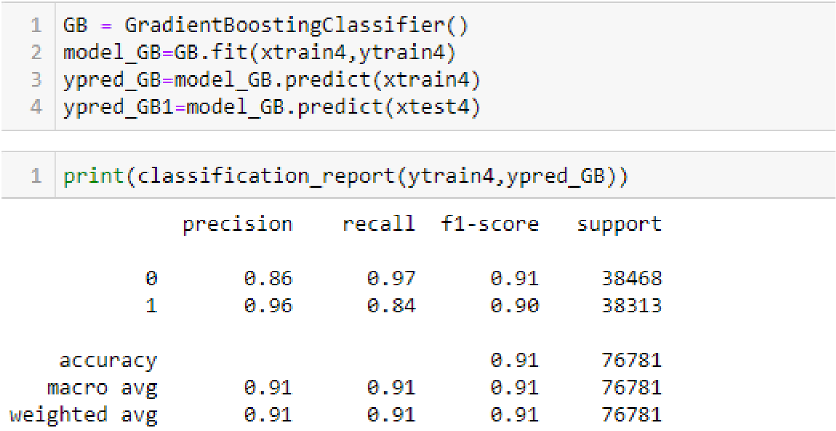
```
AD = AdaBoostClassifier()
model_AD=AD.fit(xtrain4,ytrain4)
ypred_AD=model_AD.predict(xtrain4)
ypred_AD=model_AD.predict(xtest4)
```

```
print(classification_report(ytest4,ypred_AD))
```

```
              precision    recall  f1-score   support

           0       0.75      0.68      0.71     16376
           1       0.71      0.78      0.74     16531

    accuracy                           0.73     32907
   macro avg       0.73      0.73      0.73     32907
weighted avg       0.73      0.73      0.73     32907
```

Model performance is comparatively less than previous models.

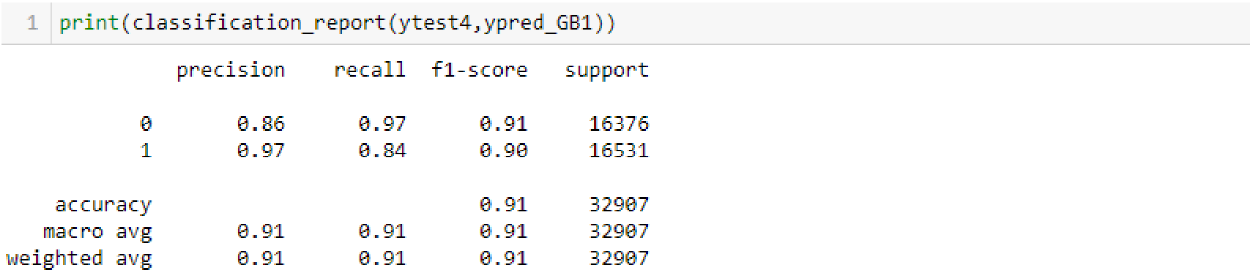
Gradient Boosting:



```
GB = GradientBoostingClassifier()
model_GB=GB.fit(xtrain4,ytrain4)
ypred_GB=model_GB.predict(xtrain4)
ypred_GB1=model_GB.predict(xtest4)
```

```
print(classification_report(ytrain4,ypred_GB))
```

```
              precision    recall  f1-score   support

           0       0.86      0.97      0.91     38468
           1       0.96      0.84      0.90     38313

    accuracy                           0.91     76781
   macro avg       0.91      0.91      0.91     76781
weighted avg       0.91      0.91      0.91     76781
```

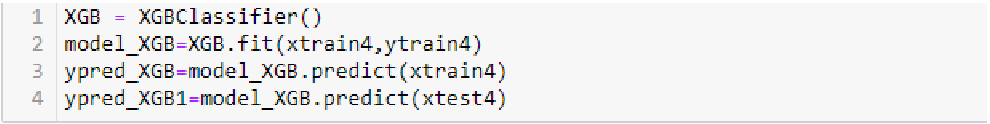


```
print(classification_report(ytest4,ypred_GB1))
```

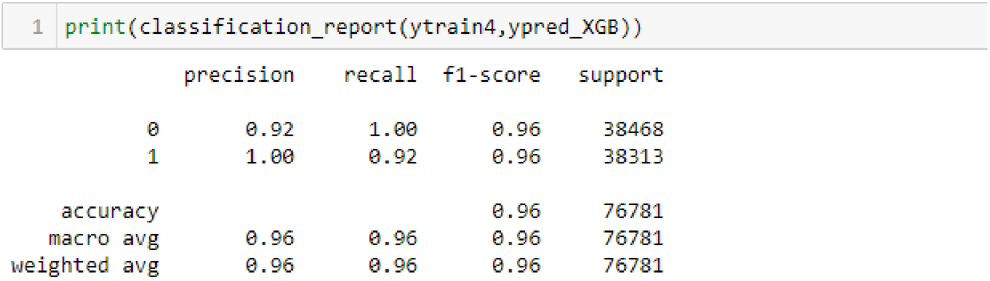
```
              precision    recall  f1-score   support

           0       0.86      0.97      0.91     16376
           1       0.97      0.84      0.90     16531

    accuracy                           0.91     32907
   macro avg       0.91      0.91      0.91     32907
weighted avg       0.91      0.91      0.91     32907
```

Model performance is good and it is better than all previous models when it comes to generalization but the recall score in in 84% which suggested around 16% of False Negatives.

Extreme Gradient Boosting:

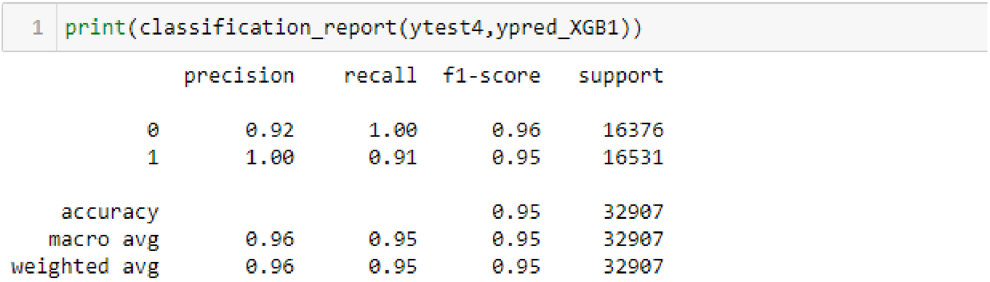


```
XGB = XGBClassifier()
model_XGB=XGB.fit(xtrain4,ytrain4)
ypred_XGB=model_XGB.predict(xtrain4)
ypred_XGB1=model_XGB.predict(xtest4)
```



```
print(classification_report(ytrain4,ypred_XGB))
```

```
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     38468
           1       1.00      0.92      0.96     38313

    accuracy                           0.96     76781
   macro avg       0.96      0.96      0.96     76781
weighted avg       0.96      0.96      0.96     76781
```



```
print(classification_report(ytest4,ypred_XGB1))
```

```
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     16376
           1       1.00      0.91      0.95     16531

    accuracy                           0.95     32907
   macro avg       0.96      0.95      0.95     32907
weighted avg       0.96      0.95      0.95     32907
```

The above model is comparatively better than all the previous models in terms of generalization and metrics.

It has better F1 score and recall when compared to Gradient Boosting which suggests the more precision in predicting our target variable.

CONCLUSION:

The model has a high level of accuracy with an overall accuracy score of 0.95. The precision scores for both classes are high, with class 0 having a precision score of 0.92 and class 1 having a precision score of 1.00. The recall score for class 0 is also high at 1.00, indicating that the model correctly identified all instances of class 0. The recall score for class 1 is slightly lower at 0.91, indicating that the model correctly identified 91% of instances of class 1.

The F1-score is a measure that takes into account both precision and recall, and it indicates the overall performance of the model. The F1-score for both classes is high, with class 0 having an F1-score of 0.96 and class 1 having an F1-score of 0.95.

Based on these metrics, it appears that the model is performing well for insurance claim prediction.

Final XGB Model can help the insurance company automate and streamline their claims processing operations, resulting in faster, more accurate, and cost-effective claims management. Predictive models can be used to estimate the likelihood of future claims and to determine the appropriate reserves and premiums to charge.