**A Group Project Report on**

**VEHICLE INSURANCE MODEL TRAINING AND EVALUATION**

**Submitted to partial fulfillment of the academic requirements of**

**Jawaharlal Nehru Technological University Hyderabad**

**For the award of the degree of**

**Bachelor of Technology**

**in**

**(2018 – 2022)**

**Submitted By**

**T. MADHUKAR V. BHARATH Y. VISHAL**

**18311A0550 18311A0553 18311A0557**

**Under the esteemed Guidance of**

**Ms. D. SRILATHA**

**Internal Guide**

**Assistant Professor, Department of CSE**

**and**

**Mr. DEVAVARAPU SREENIVASARAO**

**Project Coordinator**

**Assistant Professor, Department of CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

***(Autonomous)***

**Yamnampet, Ghatkesar Mandal, Medchal District, Hyderabad - 501301**

**Submission Date : August 2021**

**Batch No: A-18-GP-14**



**CERTIFICATE**

This is to certify that this Group Project report on “**VEHICLE INSURANCE MODEL TRAINING AND EVALUATION**”, submitted by T.MADHUKAR (18311A0550), V.BHARATH (18311A0553) and Y.VISHAL (18311A0557) in the year 2021 in partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide work that has been carried out by them as part of their **Group Project** during **Third Year Second Semester** ,under our guidance. This report has not been submitted to any other institute or university for the award of any degree.

**Ms. D. Srilatha**

**Assistant Professor**

**Department of CSE**

**Internal Guide**

**Mr. Devavarapu Sreenivasarao Dr . Aruna Varanasi**

**Assistant Professor Professor**

**Department of CSE HOD-CSE**

**Group Project Coordinator**

**External Examiner**

**Date:**

**DECLARATION**

We, **T.MADHUKAR (18311A0550), V.BHARATH (18311A0553)**and **Y.VISHAL (18311A0557)**,students of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR,** studying IIIrd year IInd semester, **COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the Group Project work, titled **“VEHICLE INSUARNCE MODEL TRAINING AND EVALUATION”** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

**ACKNOWLEDGEMENT**

We would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

We would like to express our heart-felt gratitude to our parents without whom we would not have been privileged to achieve and fulfill our dreams. We are grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably runs the institution and has had the major hand in enabling us to do this project.

We profoundly thank **Dr**.**Aruna Varanasi**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to our work.

We would like to thank our internal guide **Ms. D. Srilatha** for his/her technical guidance, constant encouragement and support in carrying out our project at college.

We also extend our gratitude to our project coordinator **Mr. Devavarapu Sreenivasarao**, who lent us valuable guidance and led us towards the staged completion of the project.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**T. MADHUKAR 18311A0550**

**V.BHARATH 18311A0553**

**Y. VISHAL 18311A0557**

**VEHICLE INSUARNCE MODEL TRAINING AND EVALUATION**

**Abstract**

Vehicle insurance is insurance for cars, trucks, motorcycles, and other road vehicles. Its primary use is to provide financial protection against physical damage or bodily injury resulting from traffic collisions and against liability that could also arise from incidents in a vehicle an insurance company that has provided health insurance to its customers, now they need in building a model to predict whether the policyholders from past year will also be interested in vehicle insurance provided by company. vehicle insurance model training and prediction can then accordingly plan communication strategy to reach out to those customers and optimize its business model and revenue. In this paper we develop the vehicle insurance model for the health insurance company based on the Synthetic Minority Over-sampling Technique (SMOTE) analysis and classification techniques. The proposed framework aims to minimize the human intervention, the obtained results reveal the high-performance gain achieved by classifying the customers based upon the response.

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Figure No.** | **Figure Title** | **Page No.** |
| **1** | **1.1** | Proposed System | **2** |
| **2** | **3.1** | Architecture Design | **6** |
| **3** | **3.3.1** | Use case Diagram | **7** |
| **4** | **3.3.2** | Class diagram | **8** |
| **5** | **3.3.3** | Sequence diagram | **9** |
| **6** | **3.3.4** | Activity diagram | **10** |
| **7** | **5.1** | Displaying the data rows | **21** |
| **8** | **5.2** | Information of the dataset | **22** |
| **9** | **5.3** | Description of the dataset | **22** |
| **10** | **5.4** | Univariate analysis of Gender | **23** |
| **11** | **5.5** | Correlation with respect to Gender | **23** |
| **12** | **5.6** | Univariate analysis of vehicle age | **24** |
| **13** | **5.7** | Correlation with respect to vehicle damage | **24** |
| **14** | **5.8** | Univariate analysis of vehicle damage | **25** |
| **15** | **5.9** | Correlation with respect to vehicle age | **25** |
| **16** | **5.10** | Bi-variate analysis of a person bought health insurance vs age | **26** |
| **17** | **5.11** | Bi-variate analysis of a person does not buy health insurance vs age | **26** |
| **18** | **5.12** | Correlation with respect to age | **27** |
| **19** | **5.13** | Bi -Variate Analysis of the person bought health insurance Vs annual premium. | **27** |
| **20** | **5.14** | Bi -Variate Analysis of the person does not buy health insurance Vs annual premium. | **28** |
| **21** | **5.15** | Correlation with respect to Annual premium. | **28** |
| **22** | **5.16** | Multi Variate Analysis. | **29** |
| **23** | **5.17** | Confusion Matrix of the test data after Decision tree classifier. | **30** |
| **24** | **5.18** | Classification report of train data | **30** |
| **25** | **5.19** | Classification report of test data | **31** |
| **26** | **5.20** | Confusion Matrix of the test data after Random Forest Classifier. | **31** |
| **27** | **5.21** | Classification report of train data | **32** |
| **28** | **5.22** | Classification report of test data | **32** |
| **29** | **5.23** | Confusion Matrix of the test data after Logistic Regression. | **32** |
| **30** | **5.24** | Classification report of train data | **33** |
| **31** | **5.25** | Classification report of test data | **33** |
| **32** | **5.26** | Confusion Matrix of the test data after KNN Classifier. | **34** |
| **33** | **5.27** | Classification report of train data | **34** |
| **34** | **5.28** | Classification report of test data | **35** |
| **35** | **5.29** | Confusion Matrix of the test data after XGBoost Classifier. | **35** |
| **36** | **5.30** | Classification report of train data | **36** |
| **37** | **5.31** | Classification report of test data | **36** |
| **38** | **5.32** | Confusion Matrix of the test data after Gradient Boost Classifier. | **37** |
| **39** | **5.33** | Classification report of train data | **37** |
| **40** | **5.34** | Classification report of test data | **37** |
| **41** | **5.35** | Confusion Matrix of the test data after CategoricalNB Classifier. | **38** |
| **42** | **5.36** | Classification report of train data | **38** |
| **43** | **5.37** | Classification report of test data | **38** |
| **44** | **5.38** | Confusion Matrix of the test data after Linear SVC Classifier. | **39** |
| **45** | **5.39** | Classification report of train data | **39** |
| **46** | **5.40** | Classification report of test data | **39** |

**INDEX**

**Abstract** i

**List of Figures** ii

|  |  |  |
| --- | --- | --- |
| **S.No** | **Content** | **Page No.** |
| **1** | **INTRODUCTION** | 1. **2** |
|  | * 1. **Scope**   2. **Existing System**   3. **Proposed System** | **1**  **1**  **2** |
| **2** | **SYSTEM ANALYSIS** | **3 - 5** |
|  | **2,1. Functional Requirement Specifications**  **2.2. Performance Requirements**  **2.3. Software Requirements**  **2.4. Hardware Requirements** | **3**  **4**  **5**  **5** |
| **3** | **SYSTEM DESIGN** | **6 - 10** |
|  | **3.1. Architecture Design**  **3.2. Modules**  **3.3. UML Diagrams**  **3.3.1. Use Case Diagram**  **3.3.2. Class Diagram**  **3.3.3. Sequence Diagram**  **3.3.4. Activity Diagram** | **6**  **6**  **7**  **8**  **9**  **10** |
| **4** | **SYSTEM IMPLEMENTATION** | **11** |
| **5** | **OUTPUT SCREENS** | **21** |
| **6** | **CONCLUSION** | **40** |
| **7** | **FUTURE SCOPE** | **41** |
| **8** | **REFERENCES** | **42** |

**CHAPTER – I**

**INTRODUCTION**

**INTRODUCTION**

Health insurance is a type of insurance that covers the whole or a part of the risk of the person incurring medical expenses. It is the coverage that provides for the payments of benefits as a result of sickness or injury. It includes insurance for losses from accident, medical expenses, disability, or accidental death and dismemberment. An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee. For example, consider you may pay a premium of Rs. 5000 each year for a health insurance cover of Rs. 200,000/- so that if, God forbid, you fall ill and need to be hospitalized in that year, the insurance provider company will bear the cost of hospitalization etc. for up to Rs. 200,000. Now if you are wondering how can company bear such high hospitalization cost when it charges a premium of only Rs. 5000/-, that is where the concept of probabilities comes in picture. For example, like you, there may be 100 customers who would be paying a premium of Rs. 5000 every year, but only a few of them (say 2-3) would get hospitalized that year and not everyone. This way everyone shares the risk of everyone else.

* 1. **Scope**

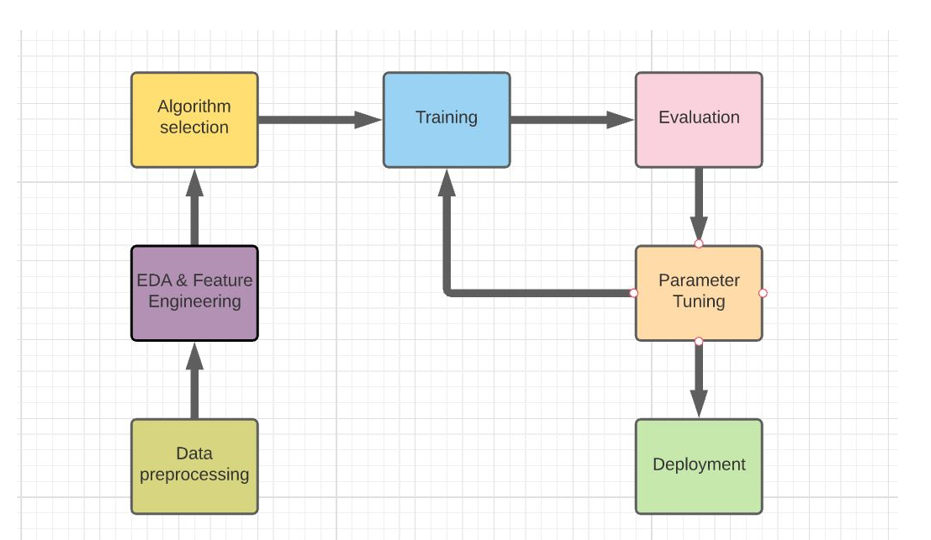
Our client is a Health Insurance company that has provided Health Insurance to its customers now they need your help in building a model to predict whether the policy-holders (customers) from past year will also be interested in vehicle insurance provided by the company. Just like medical insurance, there is vehicle insurance where every year customer needs to pay a premium of certain amount to insurance provider company so that in case of unfortunate accident by the vehicle, the insurance provider company will provide a compensation called sum assured to the customer. Building a model to predict whether a customer would be interested in Vehicle Insurance is extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers and optimize its business model and revenue. Now, in order to predict, whether the customer would be interested in Vehicle insurance, you have information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy (Premium, sourcing channel) etc.

* 1. **Existing System**

In the existing system we usually analyze few features, by various techniques, and we cannot improve the performance of the minority classes. Due to this it may cause to reduce the accuracy of our model.

* 1. **Proposed System**

In this project we discuss about Exploratory Data Analysis (EDA) for analyzing and we deal with Synthetic Minority Over-Sampling Technique (SMOTE) imbalanced classification. Imbalanced classification involves in developing the predictive models on classification datasets. The challenge of working with imbalanced datasets most machine learning models will ignore this will turn into a poor performance on the minority class, this SMOTE technique used to increase performance of the minority classes. Classification techniques used are Decision Tree classifier, Random Forest Regressor, Logistic Regression, KNN classifier, XGB classifier, Gradient Boosting Classifier, Categorical NB, Linear SVC, among above classification Model we identify the best classification Model.



**Figure 1.1**: Proposed System

**SYSTEM ANALYSIS**

This System Analysis is closely related to [requirements analysis](http://en.wikipedia.org/wiki/Requirement_analysis). It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made."This step involves [breaking down](http://en.wikipedia.org/wiki/Work_breakdown_structure) the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

**2.1 Functional Requirement Specification**

The System after careful analysis has been identified to be present with the following modules.

* 1. **User Module:**

The User is responsible for sending data to admins, view details of the policy holders a whether they are willing to take vehicle insurance.

**2. ML Operation Admin Module:**

The ML operation Admin roles are Data preprocessing which includes Data Cleaning, Data formatting, Data Sampling. Train the model using the dataset which includes the enhancement of accuracy and precision, which includes boosting and bagging technique. Deploying the Model.

**3.Server Module:**

The server module includes the storing of the categorical values, store the dataset and machine learning model, project required data from the dataset to be retrieved.

**2.2 Performance Requirements**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**2.3 Software Requirements**:

**Operating System:**  Microsoft Windows XP and later versions.

**Languages**: Python

**Tools**: Colab Jupyter Notebook

**2.4 Hardware Requirements**:

**Processor:** Intel Dual Core processor

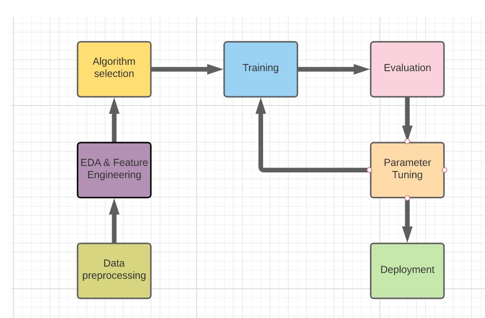
**RAM:** 512 MB

**Hard Disk:** 2 GB

**SYSTEM DESIGN**

Systems design is the process of defining the architecture, components, modules, interfaces, and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). One could see it as the application of [systems theory](http://en.wikipedia.org/wiki/Systems_theory) to [product development](http://en.wikipedia.org/wiki/Product_development). [Object-oriented analysis and design](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) methods are becoming the most widely used methods for computer systems design.

**3.1 Architectural Design**

****

**Fig 3.1** : Architectural Design

Dataset is sent from the insurance company to the ML operation admin then data preprocessing takes place after data preprocessing, then Exploratory Data analysis which include Univariate analysis, Bi- Variate analysis and feature engineering , after algorithm selection, we train the data, we evaluate the data based upon the testing dataset, if performance is low then we parameter tuning, in parameter tuning we identify different parameters that are strongly correlated with the target variable then deployment of the model takes place.

* 1. **Modules**

**1.User Module:**

The User is responsible for sending data to admins, view details of the policy holders a whether they are willing to take vehicle insurance.

**2. ML Operation Admin Module:**

The ML operation Admin roles are Data preprocessing which includes Data Cleaning, Data formatting, Data Sampling. Train the model using the dataset which includes the enhancement of accuracy and precision, which includes boosting and bagging technique. Deploying the Model.

**3.Server Module:**

The server module includes the storing of the categorical values, store the dataset and machine learning model, project required data from the dataset to be retrieved.

**3.3 UML Diagrams**

UML Diagrams for our application are as follows:

**3.3.1 Use Case Diagrams**

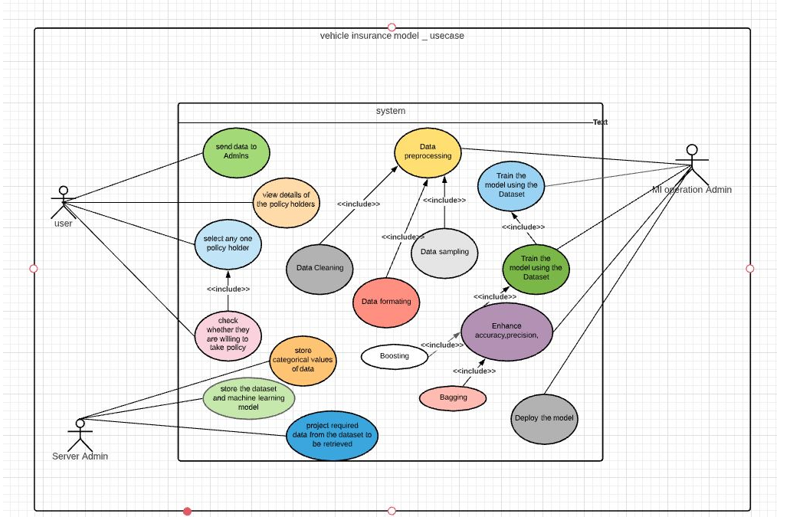
****

Fig 3.1.1: Use case Diagram for entire application

**User Use Case:**

The User is responsible for sending data to admins, view details of the policy holders a whether they are willing to take vehicle insurance.

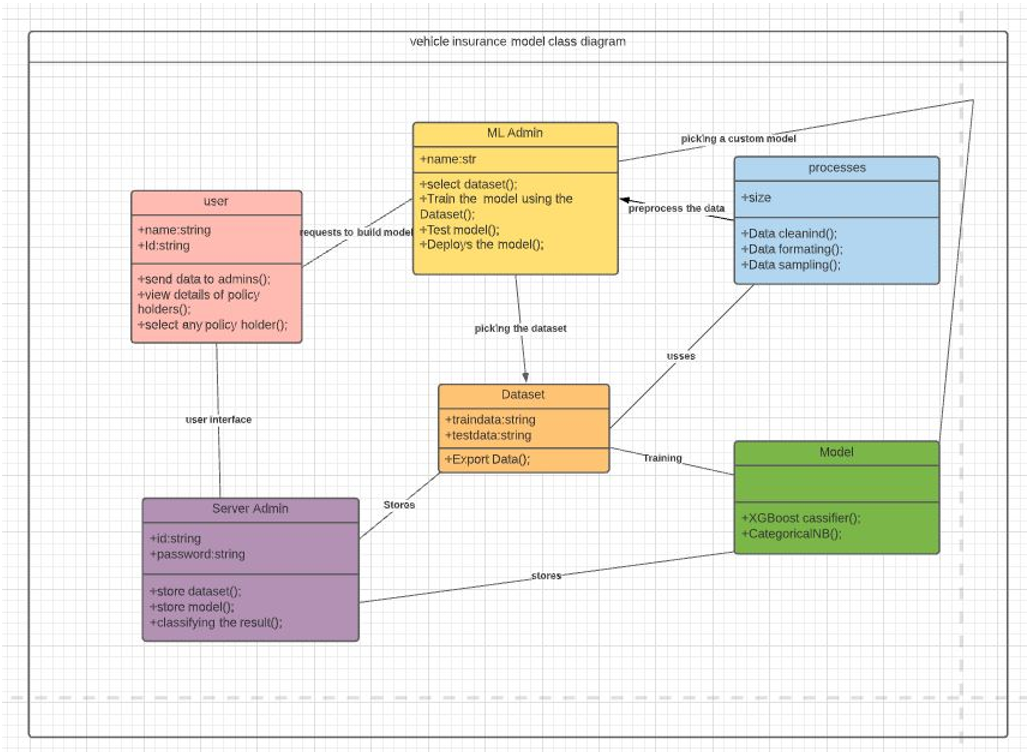
**ML Operation Admin Use Case:**

The ML operation Admin roles are Data preprocessing which includes Data Cleaning, Data formatting, Data Sampling. Train the model using the dataset which includes the enhancement of accuracy and precision, which includes boosting and bagging technique. Deploying the Model.

**Server Module Use Case:**

The server module includes the storing of the categorical values, store the dataset and machine learning model, project required data from the dataset to be retrieved.

**3.3.2 Class Diagrams**

****

**Fig** 3.3.2 class Diagram for entire application

In our application we mainly identified six classes namely User, Dataset, ML admin, Server Admin, Processes, Model and we have identified attributes and operations for these classes. These classes along with their respective attributes and operations are depicted in above class diagram.

**3.3.3 Sequence Diagrams**

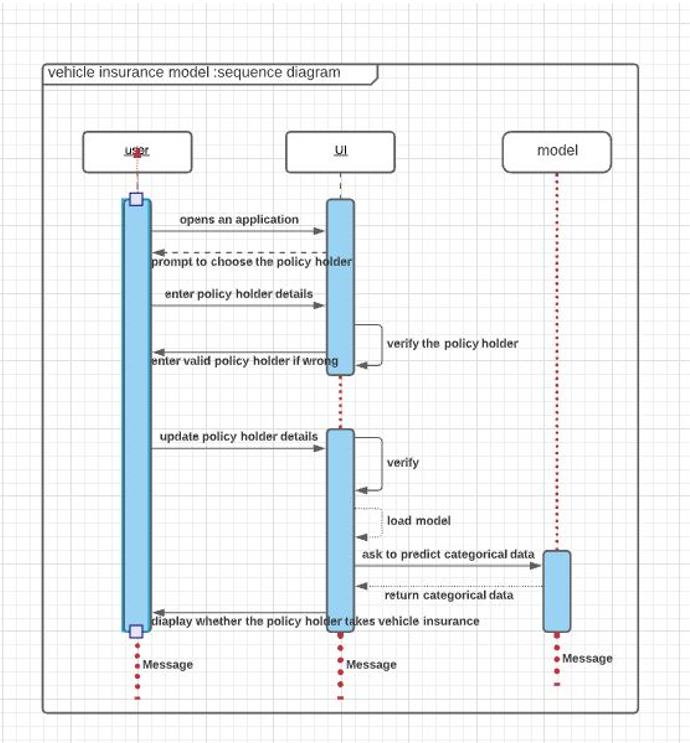
****

Fig 3.3.3: Sequence Diagram for entire application.

In Sequence diagram sequence of actions are triggered, at first user enters the application, the User Interface send to prompt to choose the policy holders to the user,

then user enter the policy holder details, then interface verify the policy holder credentials, update their results to the user, if wrong details are entered then updating the policy holder details by the user, verifying the details again, loading the model by the interface, interface ask model to predict the categorical value, and return the categorical data.

**3.3.4 Activity Diagrams**

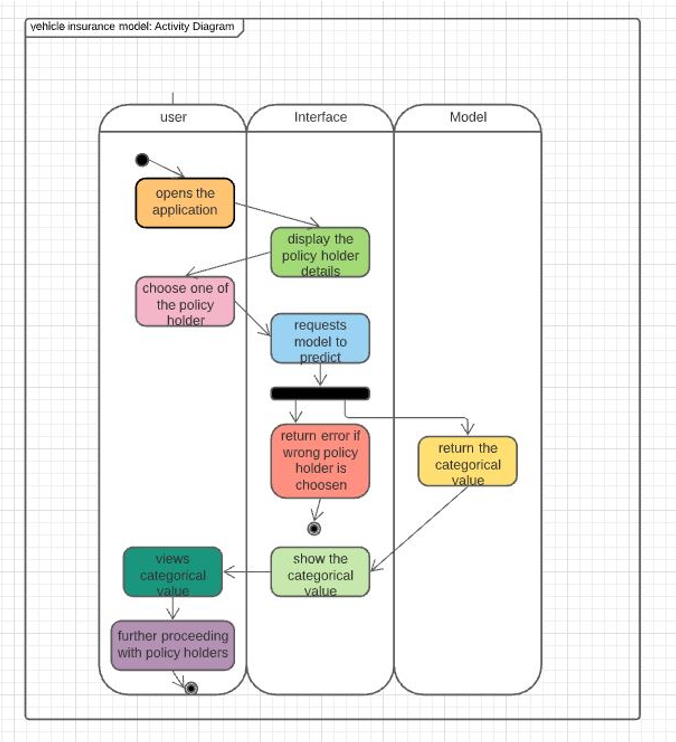
****

Fig 3.3.4: Activity diagram of entire model.

In activity diagram sequence of actions are triggered, the application is starting by the user, at first user enters the application, the User Interface send to prompt to choose the policy holders to the user, then user enter the policy holder details, then interface verify the policy holder credentials, update their results to the user, if wrong details are entered then updating the policy holder details by the user, verifying the details again, loading the model by the interface, interface ask model to predict the categorical value, and return the categorical data, the application end.

**SYSTEM IMPLEMENTATION**

The implementation stage of any project is a true display of the defining moments that make a project a success or a failure. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

**Code:**

from google.colab import drive

drive.mount('/content/drive')

import numpy as np

from google.colab.patches import cv2\_imshow

import cv2

import matplotlib.pyplot as plt

Image=cv2.imread('/content/drive/MyDrive/frontpage.JPG')

cv2\_imshow(Image)

**Importing modules and Loading datasets**

import pandas as pd

import plotly.express as px

import matplotlib.pyplot as plt

import seaborn as sns

from imblearn.over\_sampling import SMOTE

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier,AdaBoostClassifier

from xgboost import XGBClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import classification\_report,confusion\_matrix,r2\_score

from sklearn.naive\_bayes import CategoricalNB

from sklearn.svm import LinearSVC

import warnings

warnings.filterwarnings('ignore')

reading the dataset, information regarding the dataset, displaying the first 5 tuples, describing the data.

data=pd.read\_csv('/content/drive/MyDrive/train.csv')

data.head()

data.info()

data.drop(['id'],inplace=True,axis=1)

data.describe()

**Uni-Variate anlysis:**

**Gender:**

plt.figure(figsize=(3,3))

plt.pie(data.Gender.value\_counts(),explode=[.1,.2],startangle=90,autopct='%.2f%%',colors=['#FF6103','#7FFF00'],radius=4,labels=['Male','Female'])

plt.title('Gender',fontdict={'fontsize':20,'fontweight':'bold'})

plt.axis('equal')

plt.show()

countplot:

plt.figure(figsize=(10,8))

sns.countplot(data.Gender,hue=data.Response,palette='Oranges')

plt.show()

correlation:

data.Gender=pd.Categorical(data.Gender,categories=['Male','Female'],ordered=True).codes

correlation1=data.corr()

correlation1[['Gender']].sort\_values(by='Gender',ascending=False).style.background\_gradient(cmap='Blues')

**Vehicle age:**

plt.figure(figsize=(3,3))

plt.pie(data.Vehicle\_Age.value\_counts(),explode=[.1,.1,.1],startangle=90,autopct='%.2f%%',radius=4,colors=['#FF6103','#7FFF00', '#DC143C'],labels=['1-2 years','< 1 year','> 2 years'])

plt.title('Vehical Age',fontdict={'fontsize':22,'fontweight':'bold'})

plt.axis('equal')

plt.show()

Countplot:

plt.figure(figsize=(10,8))

sns.countplot(data.Vehicle\_Age,hue=data.Response,palette='Greens')

plt.show()

Correlation:

data.Vehicle\_Age=pd.Categorical(data.Vehicle\_Age,categories=['1-2 Year','< 1 Year','> 2 Years'],ordered=True).codes

correlation2=data.corr()

correlation2[['Vehicle\_Age']].sort\_values(['Vehicle\_Age'],ascending=False).style.background\_gradient(cmap='Greys')

**Vehicle damage:**

plt.figure(figsize=(8,8))

plt.pie(data.Vehicle\_Damage.value\_counts(),explode=[.1,.1],startangle=90,autopct='%.2f%%',colors=['#1E90FF','#FFD700',],radius=4,labels=['Yes','No'])

plt.title('Vehicle Damage',fontdict={'fontsize':22,'fontweight':'bold'})

plt.axis('equal')

plt.show()

Countplot:

plt.figure(figsize=(10,8))

sns.countplot(data.Vehicle\_Damage,hue=data.Response,palette='Purples')

plt.show()

Correlation:

data.Vehicle\_Damage=pd.Categorical(data.Vehicle\_Damage,categories=['Yes','No'],ordered=True).codes

correlation3=data.corr()

correlation3[['Vehicle\_Damage']].sort\_values(by='Vehicle\_Damage',ascending=False).style.background\_gradient(cmap='cool')

**BI-Variate Analysis:**

**Age & Health Insurance**

**Person bought health insurance Vs Age:**

plt.figure(figsize=(10,8))

sns.distplot(data.Age,color='red',label='Age')

sns.distplot(data.Age[data.Response==1],color='purple')

plt.title('Person bought health insurance Vs Age',fontdict={'fontsize':20,'fontweight':'bold'})

plt.show()

**Person doesnot bought health insurance Vs Age:**

plt.figure(figsize=(10,8))

sns.distplot(data.Age,color='red',label='Age')

sns.distplot(data.Age[data.Response==0],color='blue')

plt.title('Person doesnot bought health insurance Vs Age',fontdict={'fontsize':20,'fontweight':'bold'})

plt.show()

Scaling down age:

max\_age=max(data.Age)

min\_age=min(data.Age)

data.Age=data.Age.apply(lambda x: (x-min\_age)/(max\_age-min\_age))

Correlation:

correlation=data.corr()

correlation[['Age']].sort\_values(by='Age',ascending=False).style.background\_gradient(cmap='Blues')

**Annual Premium & Health insurance:**

**Person bought health insurance Vs Annual premium:**

plt.figure(figsize=(10,8))

sns.distplot(data.Annual\_Premium,color='red',label='Annual Premium')

sns.distplot(data.Annual\_Premium[data.Response==1],color='blue')

plt.title('Person bought health insurance Vs Annual premium',fontdict={'fontsize':20,'fontweight':'bold'})

plt.show()

**Person doesnot bought health insurance Vs Health Insurance:**

plt.figure(figsize=(10,8))

sns.distplot(data.Annual\_Premium,color='red',label='Annual Premium')

sns.distplot(data.Annual\_Premium[data.Response==0],color='green')

plt.title('Person doesnot bought health insurance Vs Annual premium',fontdict={'fontsize':20,'fontweight':'bold'})

plt.show()

Scaling Down Annual Premium:

max\_premium=max(data.Annual\_Premium)

min\_premium=min(data.Annual\_Premium)

data.Annual\_Premium=data.Annual\_Premium.apply(lambda x: (x-min\_premium)/(max\_premium-min\_premium))

Correlation:

correlation[['Annual\_Premium']].sort\_values(by='Annual\_Premium',ascending=False).style.background\_gradient(cmap='Blues')

**MultiVariate Analysis:**

plt.figure(figsize=(12,8))

sns.heatmap(correlation,cmap='RdYlGn')

plt.show()

**Using SMOTE to solve the problem of imbalanced data:**

oversample=SMOTE()

X,y=oversample.fit\_resample(data.iloc[:,:10],data.iloc[:,10])

**Splitting train-test Data**

train\_x,test\_x,train\_y,test\_y=train\_test\_split(X,y,test\_size=.1,random\_state=42)

**Model Training And evaluation:**

**Decision TreeClassifier**

from sklearn.tree import DecisionTreeClassifier

model1=DecisionTreeClassifier(random\_state=42,max\_depth=8)

grid1=GridSearchCV(model1,param\_grid={'max\_depth':range(5,8)})

grid1.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier1.predict(test\_x)

pred\_train\_y=classifier1.predict(train\_x)

cm1=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm1,annot=True,cmap='RdYlGn')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of test data \n',classification\_report(test\_y,pred\_test\_y))

**RandomForest Classifier**

classifier2=RandomForestClassifier(random\_state=42,max\_depth=6)

classifier2.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier2.predict(test\_x)

pred\_train\_y=classifier2.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='PiYG')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**Logistic Regression**

classifier3=LogisticRegression(tol=0.01,max\_iter=1000)

classifier3.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier3.predict(test\_x)

pred\_train\_y=classifier3.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='Greens')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**KNN Classifier:**

classifier4=KNeighborsClassifier(n\_neighbors=100)

classifier4.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier4.predict(test\_x)

pred\_train\_y=classifier4.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='RdYlGn')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**XGBoost Classifier:**

classifier5=XGBClassifier()

classifier5.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier5.predict(test\_x)

pred\_train\_y=classifier5.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='YlGn')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**Gradient Boosting Classifier**

classifier6=GradientBoostingClassifier(random\_state=42)

classifier6.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier6.predict(test\_x)

pred\_train\_y=classifier6.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='RdYlGn')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**Categorical NB:**

classifier7=CategoricalNB()

classifier7.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier7.predict(test\_x)

pred\_train\_y=classifier7.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='RdBu')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

Linear SVC

classifier8=LinearSVC()

classifier8.fit(train\_x,train\_y)

Confusion Matrix:

pred\_test\_y=classifier8.predict(test\_x)

pred\_train\_y=classifier8.predict(train\_x)

cm2=confusion\_matrix(test\_y,pred\_test\_y)

plt.figure(figsize=(10,6))

sns.heatmap(cm2,annot=True,cmap='winter')

plt.title('Confusion metrices of test data',fontdict={'fontsize':18,'fontweight':'bold'})

plt.show()

Classification Report:

print('Classification report of train data \n',classification\_report(train\_y,pred\_train\_y))

print('Classification report of train data \n',classification\_report(test\_y,pred\_test\_y))

**5. OUTPUT SCREENS**

Output Screens of various functionalities in our application are shown over here along with the description.

After importing the data, we are displaying the top 5 rows of the dataset.

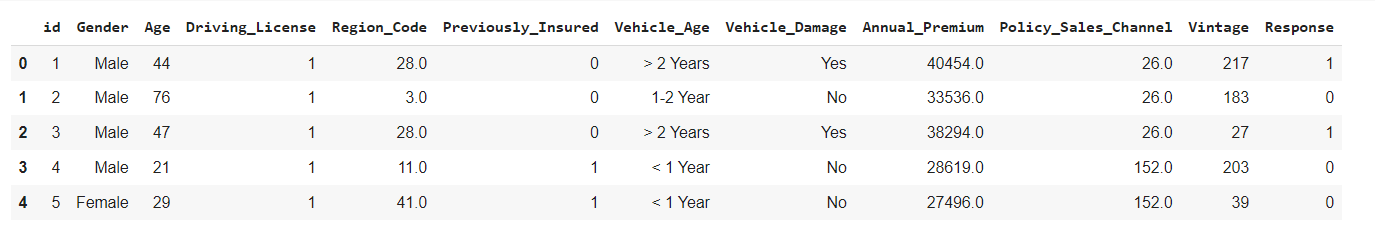


Fig 5.1 Displaying the data rows.

Information of the data, which includes the features type, features, whether not null o null.

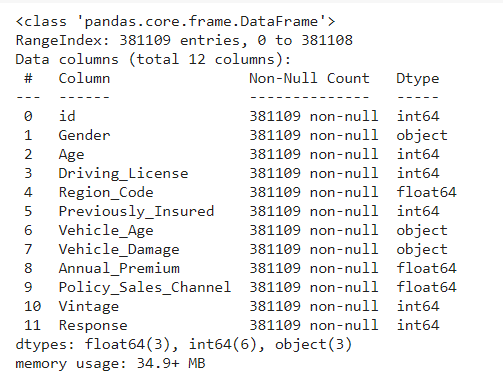


Fig 5.2 information of the dataset

Description of the data, which includes about the statistical information on the dataset, which contains the count, mean, std, min, 25%, 50%, 75% of the maximum value.

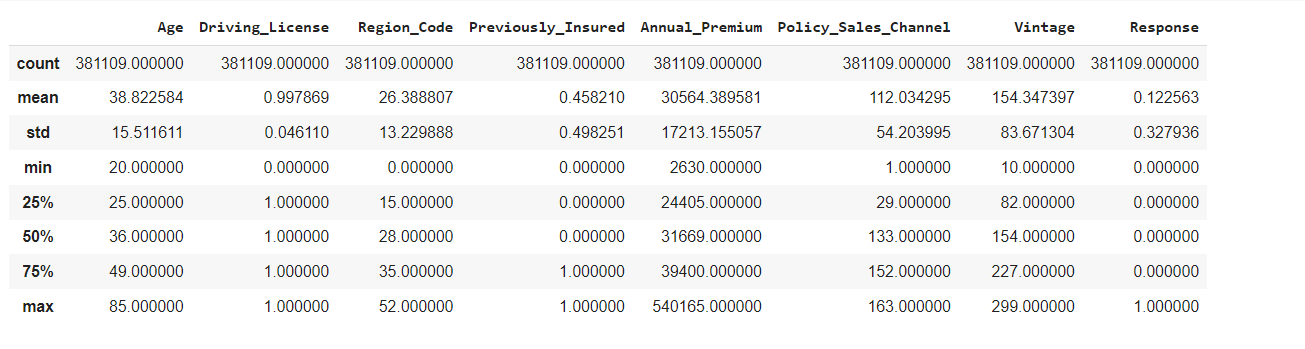
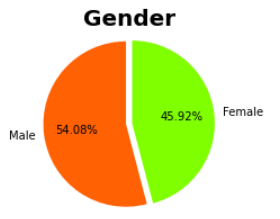


Fig 5.3 Description of the dataset

**Univariate analysis**

Uni-Variate analysis of the Gender category:



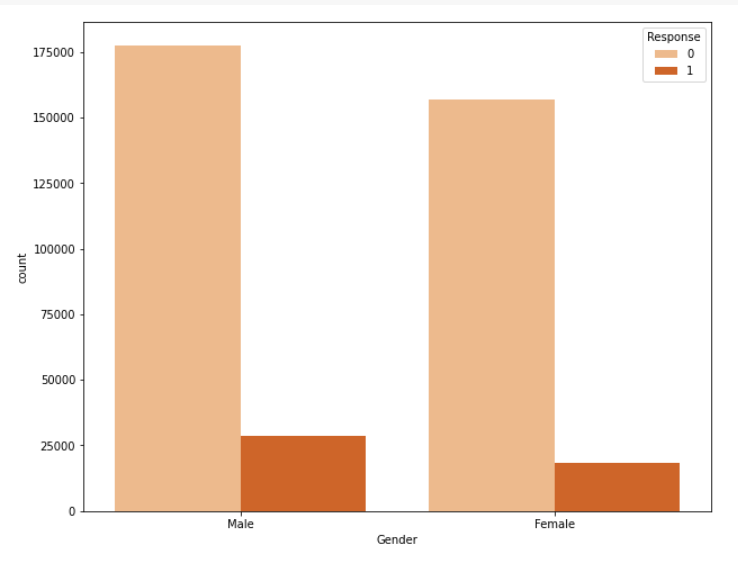


Fig 5.4 Univariate analysis of the gender.

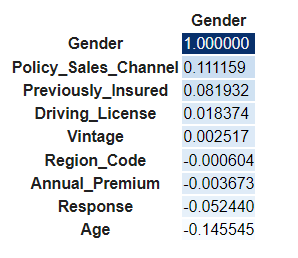
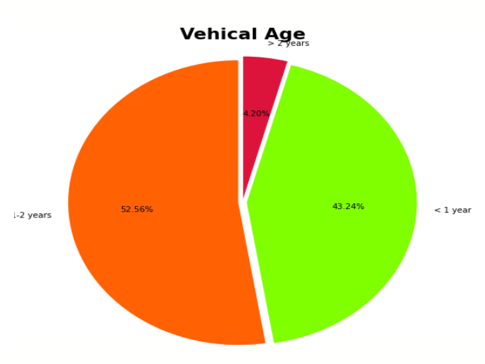


Fig 5.5 Correlation of the Gender Category In descending order

Uni- variate analysis of the Vehicle Age:



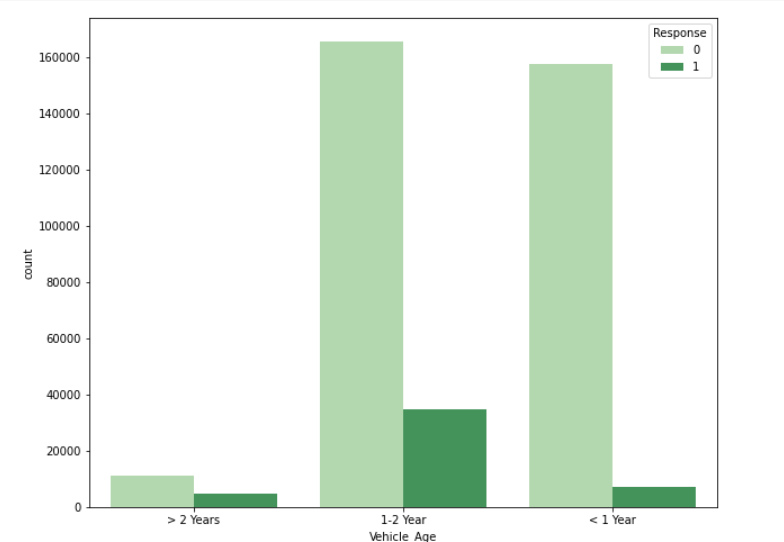


Fig : 5.6 Univariate analysis of the vehicle age

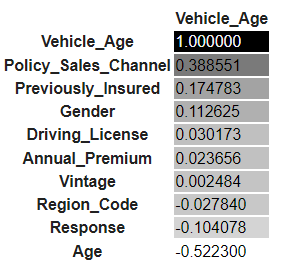
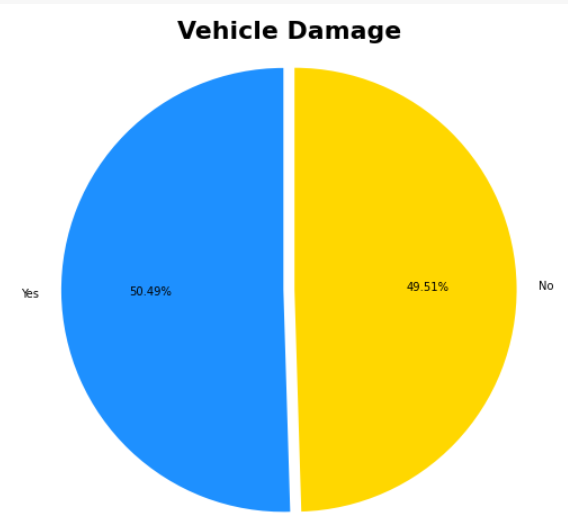


Fig 5.7 Correlation of the vehicle age in descending order

Univariate analysis of the Vehicle Damage:



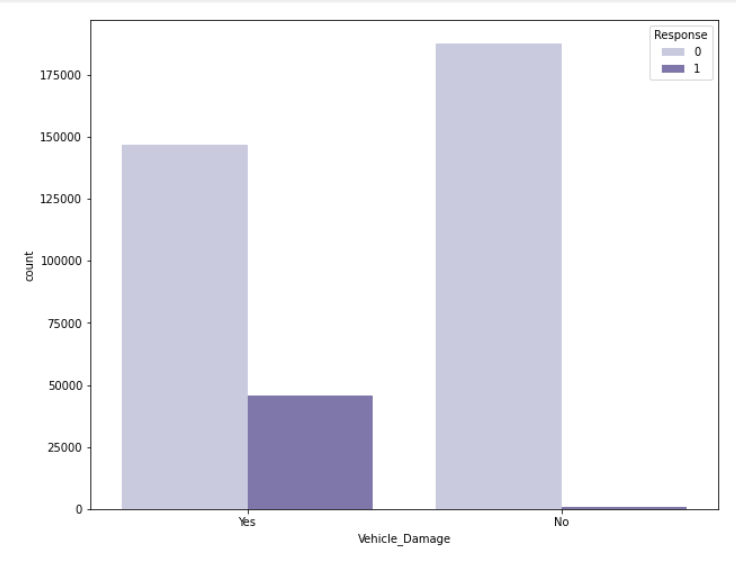


Fig :5.8 Univariate analysis of the vehicle damage.

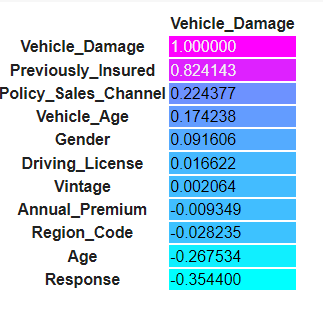


Fig 5.9 Correlation of the vehicle age in descending order

Here it is quite evident that vehicle damage is highly correlated with, the feature that was previously insured.

**Bi -Variate Analysis:**

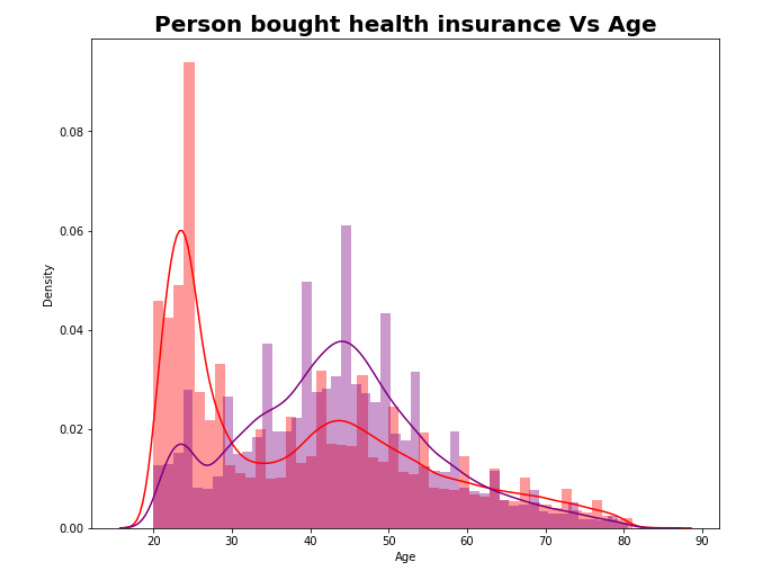
****

Fig 5.10: Bi -Variate Analysis of the person bought health insurance Vs age

We observe that age group 40 -50 have higher chance of buying the health insurance

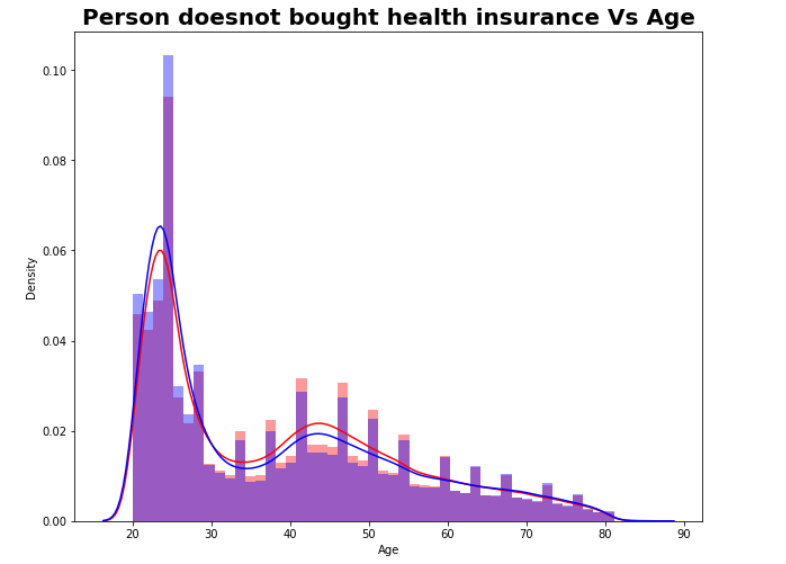


Fig 5.11: Bi -Variate Analysis of the person not bought health insurance Vs age

We observe that age group 20 -30 have lower chance of buying the health insurance.

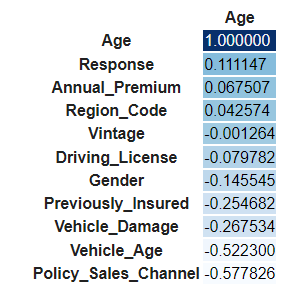


Fig 5.12: Correlation of the age factor in descending order.

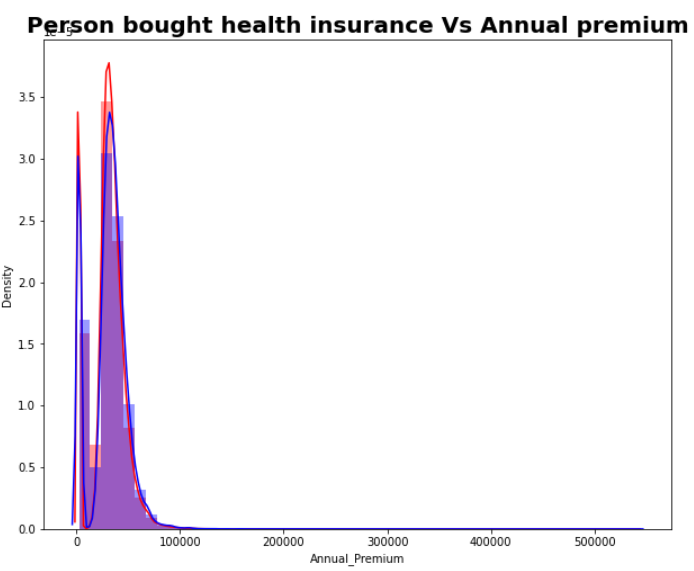


Fig 5.13: Bi -Variate Analysis of the person bought health insurance Vs annual premium.

We observe that annual premium with 30000-60000 highest chance of buying health insurance.

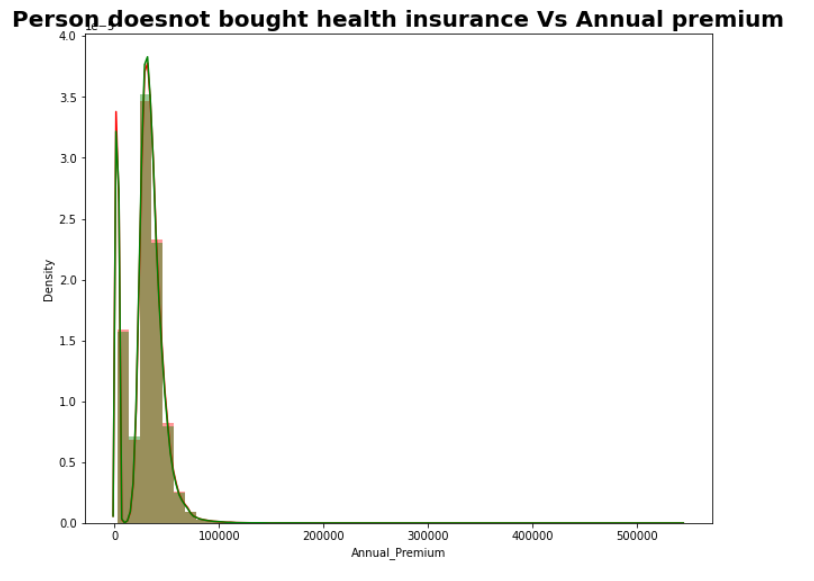


Fig 5.14: Bi -Variate Analysis of the person does not buy health insurance Vs annual premium.

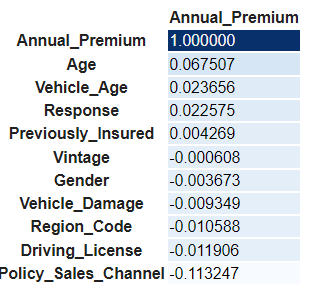


Fig 5.15: Correlation of the annual premium in descending order.

**Multi Variate Analysis:**

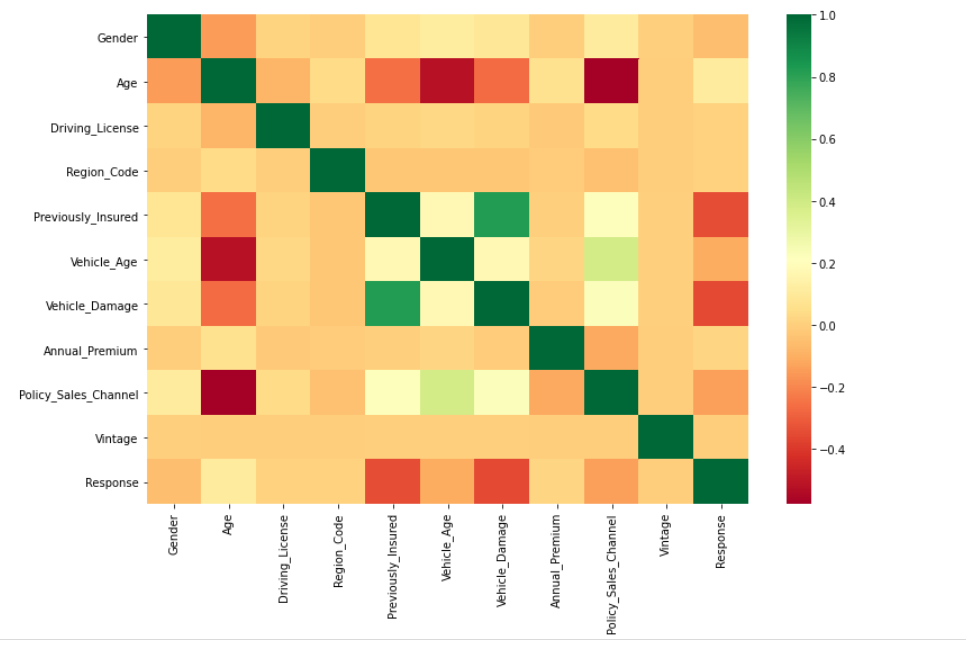
****

Fig 5.16: Multi variable analysis.

Multi variate analysis of the all the features which include correlation of the all the features. Here we observe that vehicle insurance is highly correlated with the previously insured.

**Decision Tree classifier:**

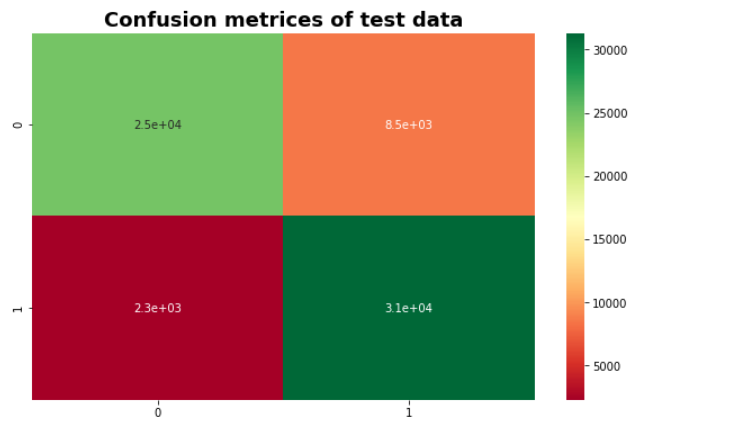
****

Fig 5.17 Confusion Matrix of the test data after Decision tree classifier.

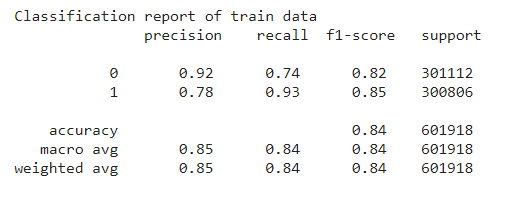


Fig 5.18 Classification report of train data.

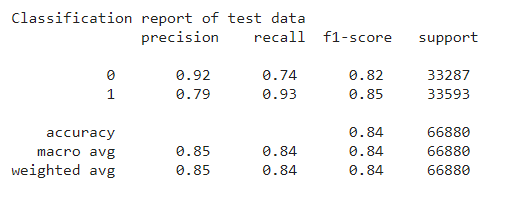


Fig 5.19: Classification report of test data

Here we observe that the accuracy by using decision tree classifier is 84% .

# RandomForest Classifier:

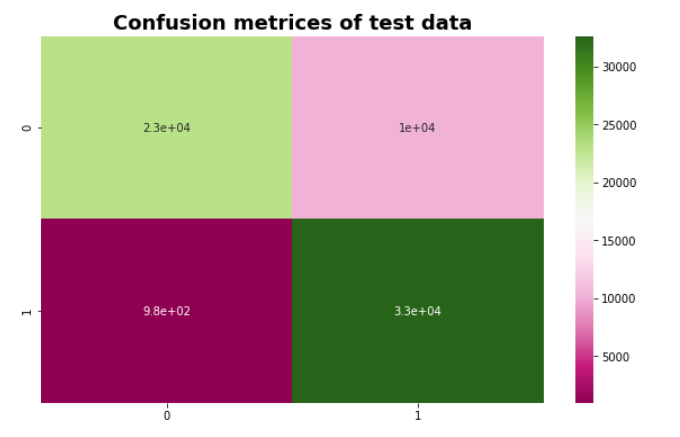


Fig 5.20 Confusion Matrix of the test data after Random Forest Classifier.

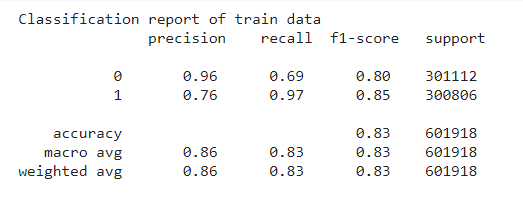


Fig 5.21 Classification report of train data.

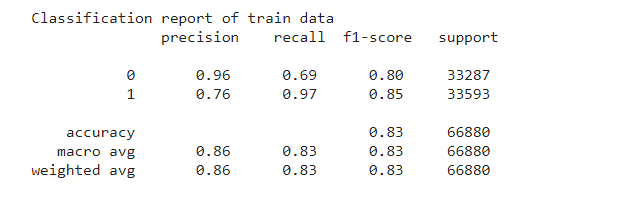


Fig 5.22 Classification report on test data.

Here we observe that the accuracy by using Random Forest classifier is 83%.

# Logistic Regression:

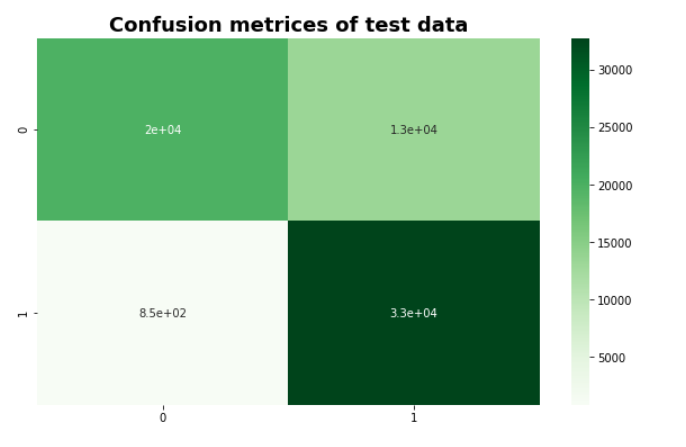


Fig 5.23: Confusion Matrix of the test data after Logistic Regression.

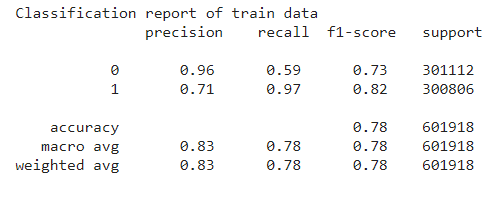


Fig 5.24 Classification report of train data.

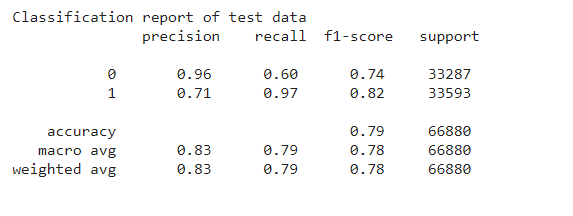


Fig 5.25 Classification report on test data.

Here we observe that the accuracy by using Logistic Regression is 79% .

# KNN Classifier

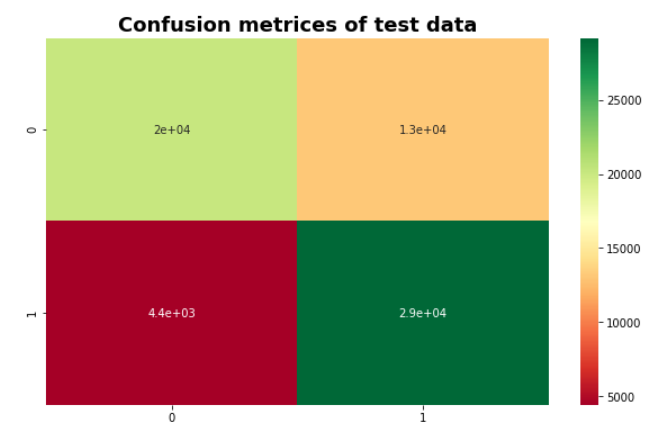


Fig 5.26: Confusion Matrix of the test data after KNN Classifier.

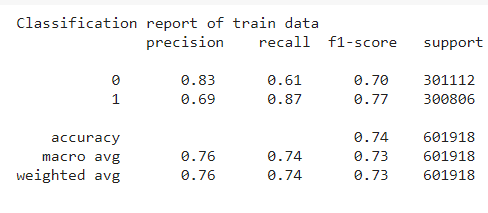


Fig 5.27 Classification report of train data.

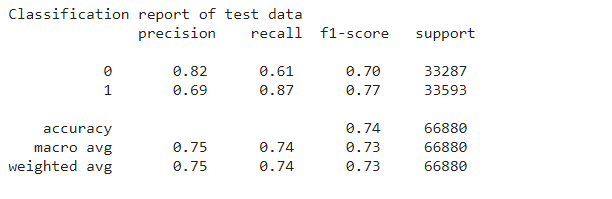


Fig 5.28 Classification report on test data.

Here we observe that the accuracy by using KNN classifier is 74% .

# XGBoost Classifier:

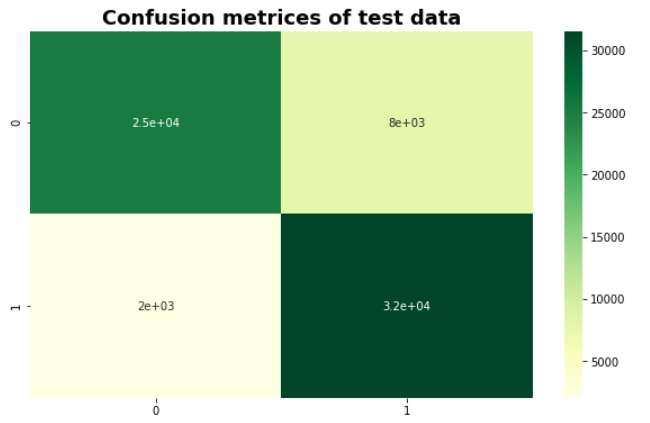


Fig 5.29: Confusion Matrix of the test data after XGBoost Classifier.

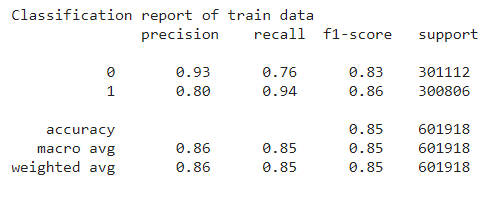


Fig 5.30 Classification report of train data.

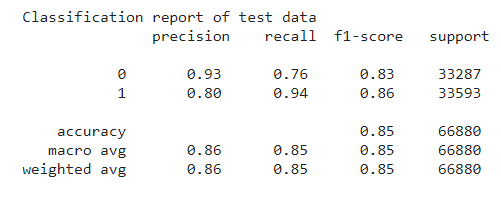


Fig 5.31 Classification report on test data.

Here we observe that the accuracy by using XGBoost classifier is 85% .

# Gradient Boosting Classifier:

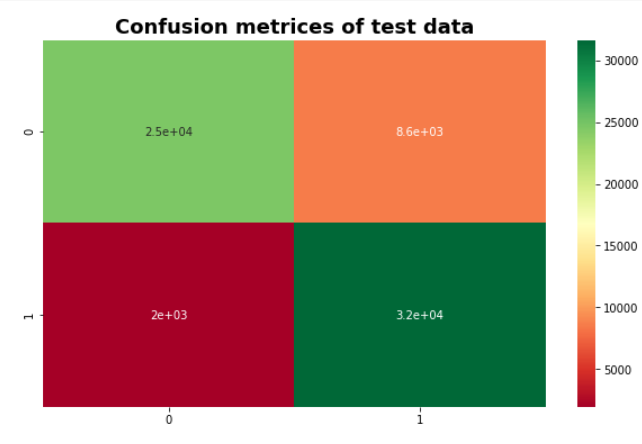


Fig 5.32 Confusion Matrix of the test data after Gradient Boost Classifier.

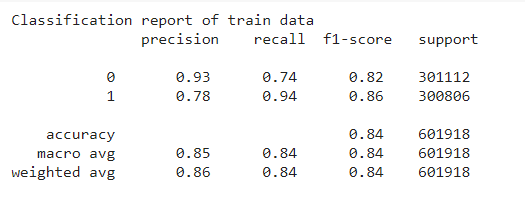


Fig 5.33 Classification report of train data.

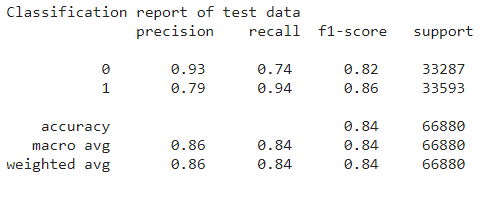


Fig 5.34 Classification report on test data.

Here we observe that the accuracy by using Gradient Boosting classifier is 84% .

# CategoricalNB

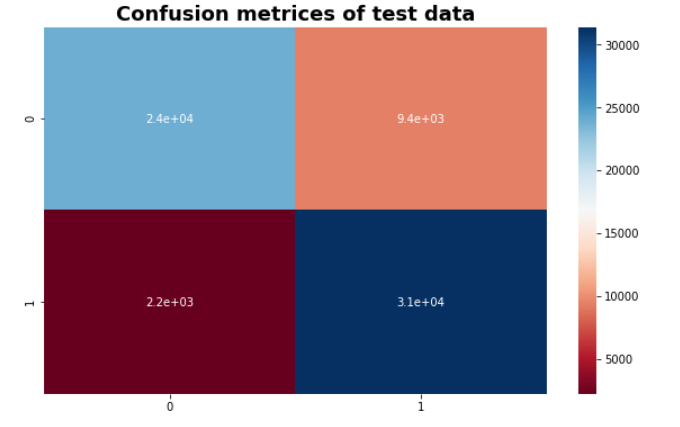


Fig 5.35 Confusion Matrix of the test data after CategoricalNB Classifier.

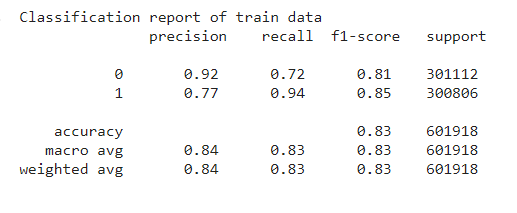


Fig 5.36 Classification report of train data.

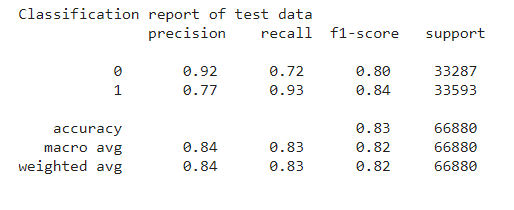


Fig 5.37 Classification report on test data.

Here we observe that the accuracy by using CategoricalNB is 83% .

# LinearSVC

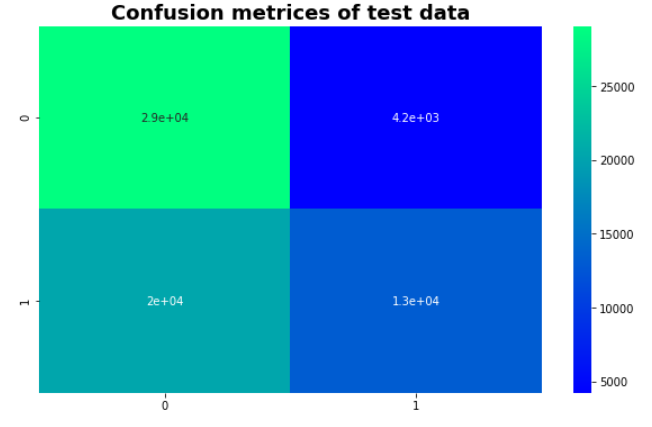


Fig 5.38 Confusion Matrix of the test data after Linear SVC Classifier.

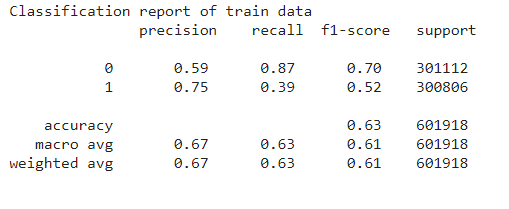
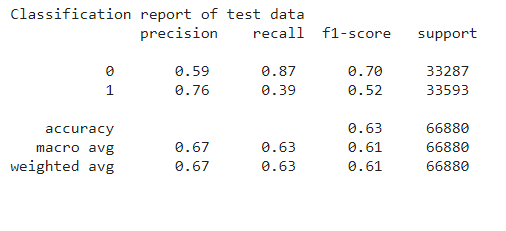


Fig 5.39 Classification report of train data.

 Fig 5.40 Classification report on test data.

Here we observe that the accuracy by using LinearSVC is 6

**CONCLUSION**

Using the Decision tree classifier, the accuracy score is 84%. Using the Random Forest Classifier, the accuracy score is 83%. Using Logistic Regression, the accuracy score is 79%. Using the KNN Classifier, the accuracy score is 74%. Using the XGBoost Classifier, the accuracy score is 85%. Using the Gradient Boosting classifier, the accuracy score is 84%. Using the Categorical NB, the accuracy score is 83%. Using the LinearSVC, the accuracy score is 63%.

In this study it is quite evident that XGBoost Classifier works better than other algorithms, with the test accuracy of about 85%. This solution presents an important asset for health insurance company for classifying the customers on the basis of vehicle insurance.

**FUTURE SCOPE**

By inclusion of new features like vehicle type, classification accuracy gets changes, we have to maintain the accuracy by the latest classification techniques. Accuracy should get if we add new features or delete new features.

With reference to new insurance schemes the vehicle insurance prediction should be done, according to government regulations, we have to predict the data.

**REFERENCES**

1. *T. Badriyah, L. Rahmaniah, and I. Syarif, “Nearest neighbour and statistics method based for detecting fraud in auto insurance,” in IEEE International Conference on Applied Engineering (ICAE’18), Batam, Indonesia, October 2018.*
2. *T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in ACM International Conference on Knowledge Discovery and Data Mining, (SIGKDD’16), San Francisco, California, USA, August, 2016.*
3. *Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, “XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud,” in IEEE International Conference on Big Data and Smart Computing (BigComp’18), Shanghai, China, January 2018.*
4. *Heterogeneous Uncertainty Sampling for Supervised Learning. Catlett, J. and Lewis, D. (1994). In Transactions of a Machine Learning 11th International Conference, pages 148-156.*
5. *J. Brownlee, How to Develop Voting Ensembles with Python, Machine Learning Mastery, Apr 16, 2020. https:// machine-learningmastery.com/voting-ensembles-with-python/ (accessed Jul. 08, 2021).*
6. *ML – Gradient Boosting, GeeksforGeeks, Aug. 25, 2020. https://www.geeksforgeeks.org/ml-gradient-boosting/ (accessed Jul. 08, 2021)*
7. *Decision Trees Algorithms | by Madhu Sanjeevi ( Mady) | Deep Math Machine learning.ai |Medium. https://medium.com/-deep-math-machine-learning-ai/chapter-4- decision-trees-algorithms-b93975f7a1f1 (accessed Jul. 08, 2021).*
8. *Rohrig, K.; Lange, B., Application of wind power prediction tools for power system operations, IEEE Power Engineering Society GeneralMeeting, 18-22 June 2006.*
9. *Sharma S, Agrawal J, Agarwal S, Sharma S, “Machine Learning Techniques for Data Mining: A Survey” ,Computational Intelligence and Computing Research (ICCIC), IEEE International Conference on 26-28 Dec. 2013 Page(s):1 - 6 ,2013.*

**APPENDIX-A: About Machine Learning techniques**

1. Data Preprocessing

Data Preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format, techniques used for Data pre-processing are

Data Cleaning

Data cleaning is the process of detecting, rectifying, or removing inaccurate and corrupted information from the dataset or database. In addition, it recognizes inaccurate or unfinished parts of data, filling the missing ones, and removing the noisy data.

Data Integration

Data integration involves combining data residing in different sources and providing users with a unified view of these all data. Data integration may involve inconsistent data and therefor needs data cleaning.

Data Transformation.

Data transformation is the process of converting data from one from to another from. Data transformation is necessary to ensure that data from one application or database is understandable to other applications and databases.

Data Reduction

Data reduction involves in reducing the number of attributes, attribute values, number of tuples.

B. EDA & Feature Engineering

Exploratory Data analysis is the technique for analyzing datasets to summarize the main characteristics. There are many tools that are useful for EDA. Typical graphical techniques used are: box plot, histogram, multi-vari chart, run chart, parse to chart, odds ratio etc. Interactive versions of these plots are Dimensionality reduction. Univariate analysis, Bi-variate analysis, multi-variate comes under the exploratory data analysis.

Uni variate analysis

Univariate analysis is the analysis of the single variable in this paper we do analysis using Uni-variate analysis.

Bi -variate analysis

Bi- variate analysis is the analysis of the two variables, in this paper we do analysis using bi-variate analysis

Multi-variate analysis

Multi variate analysis is the analysis of the more than three variables, in this paper we do analysis using multi variate analysis

SMOTE (Synthetic Minority oversampling technique)

In order to increase the performance of the minority classes we use this technique by analyzing the minority classes. Imbalanced classification technique is used to increase performance.

Feature Engineering is the process of using domain knowledge to extract features from raw data. A feature is a property shared by independent units on which analysis or prediction is to be done.

1. Algorithm Selection

Decision Tree Classifier

Decision tree is a supervised machine learning technique that can be used for classification and regression. Decision nodes are used to make decision and have multiple branches, leaf node of a decision tree is the outcome of the decisions. CART (classification and regression) algorithm is used to make decision tree. In this paper, we import sklearn.tree library to implement Decision TreeClssifier.

Random Forest classifier

Random forests or random decision fore-sts are an ensemble method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set. In this paper, we import sklearn.ensemble library to implement Random Forest classifier.

Logistic Regression Classifier:

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Logistic regression transforms its output using the logistic sigmoid function to return a probability value. In this paper, we import sklearn.linear\_model library to implement logistic regression.

XGBoost Classifier:

XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data. In this paper we import the xgboost.XGBClassifier to implement XGB classifier

KNN Classifier

It assumes the similarity between the new data and available data and place the new data into a category it present. It is a Lazy learner algorithm because it does not learn from training set, instead it stores the dataset, at the time of classification action is performed. In this paper we import sklearn.neighbours library to implement KNN

Categorical Naïve bayes Classifier:

Categorical Naïve bayes is a variant of a native bayes that follows categorical distribution of discrete features. The categories of each feature are drawn from a categorical distribution. In this paper we import sklearn.naive\_bayes library to implement CategoricalNB.

Gradient Boosting Classifier.

Gradient boosting classifiers are a group of machine learning algorithms that are combine many weak learning models together to create a strong predictive model using Gradient descent function. In this paper we import sklearn.ensemble library to implement Gradient boosting classifier.

Linear SVC

Linear SVC is to fit the data you provide returning a best fit hyperplane that categorizes the data. After getting a hyperplane you can feed some features to your classifier. In this paper we import sklearn.svm to implement Linear SVC.

Confusion matrix

Confusion matrix is used to describe the performance of the classification model. Confused matrix between actual and predicted values

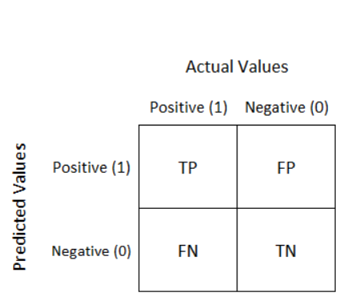


Fig 2. Confusion matrix between actual and predicted values.

|  |  |  |  |
| --- | --- | --- | --- |
| **Team No: A-18-GP-14** | | **Title** | **Guide Name** |
| **Roll No** | **Name** |
| 18311A0550 | T. MADHUKAR | **VEHICLE INSURANCE MODEL TRAINING AND EVALUATION** | Ms. D. SRILATHA  Assistant Professor |
| 18311A0553 | V.BHARATH |
| 18311A0557 | Y. VISHAL |

**ABSTRACT**

Vehicle insurance is insurance for cars, trucks, motorcycles, and other road vehicles. Its primary use is to provide financial protection against physical damage or bodily injury resulting from traffic collisions and against liability that could also arise from incidents in a vehicle an insurance company that has provided health insurance to its customers, now they need in building a model to predict whether the policyholders from past year will also be interested in vehicle insurance provided by company. vehicle insurance model training and prediction can then accordingly plan communication strategy to reach out to those customers and optimize its business model and revenue. In this paper we develop the vehicle insurance model for the health insurance company based on the Synthetic Minority Over-sampling Technique (SMOTE) analysis and classification techniques. The proposed framework aims to minimize the human intervention, the obtained results reveal the high-performance gain achieved by XGBoost in classifying the customers based upon the response.

**Student 1 Guide HOD**

**Student 2**

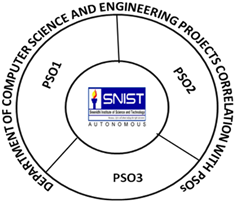
**Student 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team No: A-18-GP-14** | | **Title** | **Guide Name** |
| **Roll No** | **Name** |
| 18311A0550 | T. MADHUKAR | **VEHICLE INSURANCE MODEL TRAINING AND EVALUATION** | Ms. D. SRILATHA  Assistant Professor |
| 18311A0553 | V. BHARATH |
| 18311A0557 | Y. VISHAL |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Batch No.** | **Roll No.** | **Product/app** | **Ethics** | **research** | **social science** | **safety** |
|
| **A-18-GP-14** | 18311A0550 |  |  |  |  |  |
| 18311A0553 |
| 18311A0557 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**  snist autonomous logo  **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  **Projects Correlation with POs** | | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| **H** | **H** | **M** | **L** | **H** | **H** | **M** | **L** | **H** | **M** | **L** | **H** |

|  |  |
| --- | --- |
| **H** | **High** |
| **M** | **Moderate** |
| **L** | **Low** |

**** 

**Guide HOD**

Ms. D. Srilatha