INSURANCE FRAUD CLAIM DETECTION USING MACHINE LEARNING

SRIDHAR GUVVALA

Final Dissertation

June 2022

DEDICATION

I am thankful to my mentor who motivated to take this research study and guided me throughout the journey of this thesis paper. Thereby I dedicate this to my mentor, fellow colleagues and friends those helped me and special thanks to the University for providing this opportunity to enhance my skills and made me learn many new things throughout the entire journey.

ACKNOWLEDGEMENTS

I would like to extend my gratitude to the mentors who guided me during the research conducted. Also to all the professors of university who thought me various subjects during my tenure of study where I learnt all the topics and methodologies used to carry out the research, analyzing the data, building the model and analyzing. Which is not possible without their teaching. I would like to thank my university for giving me such a wonderful opportunity to work with and getting the all experts on board to teach the subjects.

TABLE OF CONTENT

ABSTRACT

Insurance fraud is a dissimulation or cunning act made by an individual thoughtfully to get benefited financially. Globally the cost of these frauds is billions of dollars per year combined of various insurance industries. The kind of frauds might occur in this industry are like over-inflating the claim amount, fabricating a fake scene or evidences to claim which are not entitled and hiding their faults for the occurrence of event. This causes a huge loss to the insurance provider. Generally to avoid this loss industries tend to apply rigorous and various standard methods like auditing to determine whether the claim is genuine or fake, which involves lot of human efforts, time and cost to the firm. To avoid this industries can adopt to the advanced and improving technologies like Data Science & Machine Learning for predicting the type of claim accurately within less time and cost to the company. In this research we are working on the same for automobile insurance industry, where we are doing feature engineering, build ML models using different algorithms to find the type of claim. Test those models, find the best model and do Hyper Parameter Tuning to increase the accuracy of the best model and save it as a .pkl file which is ready to be deployed on web for usage.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

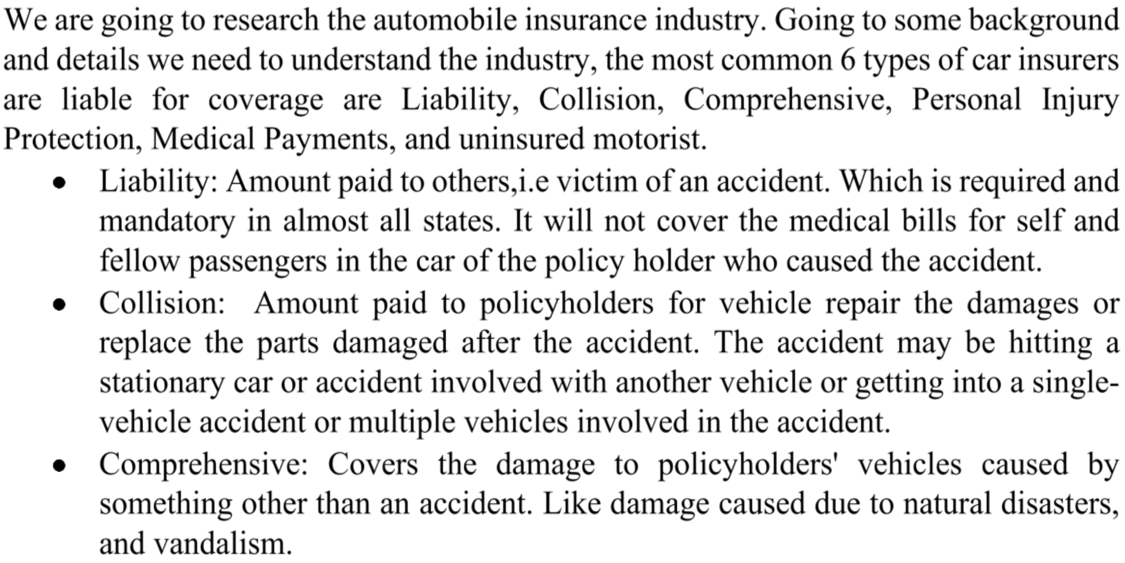
Insurance fraud isn’t a new thing, many kinds of research had been made, to determine the causes which take us back to 1945 and state these occur majorly for the financial benefit. Globally the number of fraudulent insurance claims are increased, and many cases of these are being reported. There are many types of research about these fraud cases, types of frauds, etc. occurring in various domains, and reported that the cost of insurance fraud in America is estimated to be $80 billion per annum and in developing countries like South Africa, it is $600 billion per annum. Due to this, there is an increase in the cost of insurance premiums.

The common examples of fraud cases that are reported especially in the Automobile industry are, faked accidents, over-inflated claims, claiming the amount by hiding their fault, where they will not be entitled as per the company’s terms and conditions for instance:

1) If the person does not follow the traffic rules and caused an accident, then he will not get the amount from the insurance company since it is his fault for not abiding by the rules. Though the person claims the amount by hiding or recreating a scene that there is no fault of his.
2) If the person had an accident when he is drunk, then there is no right to claim insurance. But the person claims the insurance hiding that he was drunk when the accident occurred.
3) Few people knowingly make some damage to the vehicle to claim the more amount than actual required for repair.

The Fraud Claim: It is a false claim by the user even if they don’t meet the terms of the company but will show as the Genuine and make claim to the company that they followed all their terms and is eligible for a claim for their benefits / to not occur loss.

1.1 Background of the study



We are going to research the automobile insurance industry. Going to some background and details we need to understand the industry, the most common 6 types of car insurers are liable for coverage are Liability, Collision, Comprehensive, Personal Injury Protection, Medical Payments, and uninsured motorist.

- Liability: Amount paid to others,i.e victim of an accident. Which is required and mandatory in almost all states. It will not cover the medical bills for self and fellow passengers in the car of the policy holder who caused the accident.
- Collision: Amount paid to policyholders for vehicle repair the damages or replace the parts damaged after the accident. The accident may be hitting a stationary car or accident involved with another vehicle or getting into a single-vehicle accident or multiple vehicles involved in the accident.
- Comprehensive: Covers the damage to policyholders' vehicles caused by something other than an accident. Like damage caused due to natural disasters, and vandalism.

- Personal Injury Protection: Amount paid to the policy holder's direct and indirect medical expenses after an accident. This applies to the policy holder and fellow passengers in the vehicle at the time of the accident.
- Medical Payments: Amount paid to the policy holder's medical expenses after a direct accident.
- Uninsured Motorist: Covers the damage to the vehicle and medical expenses after an accident with an uninsured driver.

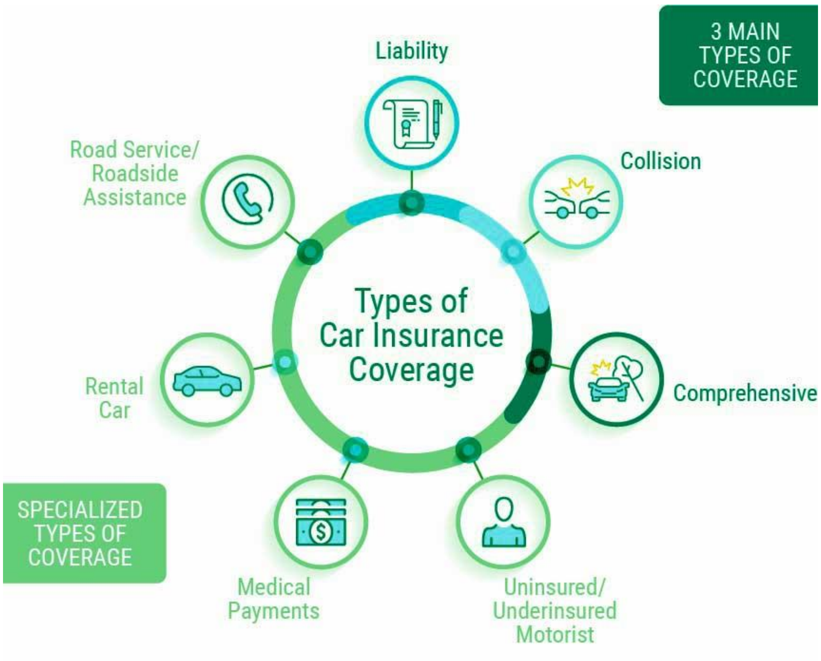

3 MAIN TYPES OF COVERAGE
Liability
Road Service/ Roadside Assistance
Collision
Types of Car Insurance Coverage
Rental Car
Comprehensive
SPECIALIZED TYPES OF COVERAGE
Medical Payments
Uninsured/ Underinsured Motorist

Fig. 1.1.1: Types of car insurance coverage

There are many other insurance types and aspects covered than the above stated. Like gap insurance, Non-owner car insurance, Mechanical breakdown insurance, etc.

General procedure followed by the industries to determine the claim: Insurance providers take many precautions and follow several procedures to determine whether the claim made is genuine or fraud for avoiding losses. In general, the methodology they follow involves a lot of human effort and intervention, where their executive is sent to the place of accident/ event, that occurred as stated by the person who reported and claimed. Then the executive will scrutinize the location, damage to the vehicle, and other things as per their company checklist or procedures to determine the claim. All this process takes a lot of effort, time, and expense for the company though failing to determine whether the claim is genuine or fraudulent and occurring more loss. Also, the long-time taken by the company to determine and release funds make the customer opt for other companies providing quick support.

1.2 Problem statement

The problem that has to be addressed is:
There is an increase in the cost of insurance globally and it is compounded by the advent of fraud, of which insurance claims fraud makes up a substantial portion.

These fraudulent claims are a big problem for the companies providing insurance.

1.3 Aim and objective

We aim to find an efficient way to determine the type of claim whether genuine or fraudulent by using the advancement of technology like applying Data Science and Machine Learning concepts since there will be huge data with the companies already. We can use that data for analyzing and building our model. Such that it will be cost-efficient to the company also they can react quickly so they can retent their customers.

- Detrmine the type of claim – Guneuine/fradulant

1.4 Scope of the study

In the Insurance industry, there are different verticals like Automobile, Life insurance, etc. as shown below in image 1.4.1. Each industry has many sub-components and a different variety of approaches/methods to be followed to decide the type of claim.

Fig. 1.4.1: Different types of Insurance

However, we are now researching the Automobile insurance industry. In the automobile industry also there is a variety of policies, terms, etc. In general, the different aspects an automobile insurance covers are as shown in the image below 1.1.1. We are utilizing the old existing data of USA records to conduct this research, limited to general overall aspects of this industry.

1.5 Structure of the study

For the above problem statement, we had a breakdown of the steps in approaching to solve the problem as below:

1.5.1 Conducting the literature review

A literature review was conducted to understand more details about the insurance industry, how it works and what kind of data is required for our study. Further, it was extended to what the fraud is and how various frauds occur in various industries also see how others conducted research in this domain, what approach and methodologies they followed to tackle this problem, and provided the solution. We then made a list of techniques followed or used by others and companies using the old and advanced techniques like big data, data science, and machine learning concepts.

1.5.2 Developing a research use case

Then we developed a use case to define the process that is to be followed such that we can successfully test a possible solution. In this section, we described all the requirements, like hardware, software, steps involved, tasks importance, outcomes of each step, and important factors to be considered for predicting insurance claims fraud. Based on this we will develop a machine learning model to predict fraud claims which are our main focus in this research.

1.5.3 Designing and developing solution

This is the method we followed to develop the solution, the flow of work we done order of work processed, and design a solution based on the research by designing the required framework, architecture and possible output to be depicted by the model. Since we are going to predict whether the claim is fraudulent or not, it comes under classification model. The set of models we are going to build for prediciton are:

- Logistic Regression
- Decession-Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- K Neighbors Classifier
- SVC

.

1.5.4 Evaluating the solution

Finally, once the models were developed by using various machine learning algorithms, we test those models with a few examples or test cases and predict the output along with the metrics such that it describes the efficiency of the models. Then we select the best model.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The main objective of this research is to develop a better and more intelligent way to predict insurance claim fraud. Our research is narrowed down to particular insurance industry. We chose the automobile industry to conduct the research and develop an intelligent way using the advancements of technology, especially leveraging the advancements in Data Science and Machine Learning. Therefore, it is required for us to know and understand the insurance industry. There is no single definition of the word 'insurance' that exists yet. In general, one of the several meanings for insurance is 'a means of guaranteeing protection or safety' (as per Merriam - Webster). Insurance is a structured, regulated method of splitting costs for detrimental experiences, therefore based on the chance or risk that a person, enterprise, or tangible resource will have injurious harm come upon them. So, if we speak about this relating to the automobile it will be the insurance provided for vehicle repair, personal medical expenses, etc. in cases of an accident occurring. The advantage of having insurance is to reduce the effect of the damage after the harm has occurred. Based on the insurance plan and the sum amount we insured, the Insurance Company will provide that amount in case any harm occurred. Modern insurance practices are arranged into a formal process for dispersing costs among people and organizations if the risk comes to fruition. Ideally, the loss is provided for in advance, by charging a premium. In this chapter, we discuss the current scenario of the automobile insurance industry from our research perspective to provide a better understanding of what is required due to the issues in the industry. In particular, the research focus is mainly on using the existing data and finding the required factors or features in data that are important to predict whether the claim is fraud or not. Keeping in mind that insurance industries have huge data available, as of research methodology we will use a dataset made for research purposes which are collected and stored as one file from various insurance companies. We will build a Machine Learning model using this data which can be used by anyone globally and predict the claim. This indeed required such that companies will be prevented from loss of fraud claim, reduce their spending on determining the claim.

2.2 Automobile insurance industry

Automobile Insurance Industry is one of the several other insurance industries. In this industry, the user will do insurance for his vehicle. The vehicle may be bikes, cars, trucks, etc. Based on the vehicle, vehicle cost, and the option chosen to get covered in case of an accident, the premium amount will vary. Also, they need to reinsure their vehicle every year, while reinsuring the cost of the vehicle will be depreciated as per norms and the sum valued for insurance will be the same. Therefore the premium amount to be paid also decreases year by year. So that the insurer will get benefited from financial loss in case of an accident occurred. Since the insurance company will bear the insured amount as per the policy the user opted. The various things that will be covered in automobile insurance areas are discussed above in chapter 1.1.

2.3 Auto insurance coverage

There are many things covered by an auto insurance policy as discussed in chapter 1.1. It will cover you and other family members on the policy, whether you are driving the car or some other person with your consent. It also covers personal driving while commuting to work, running errands, or taking a trip. Personal auto insurance will not cover if you used it for commercial purposes. In case of commercial purpose, for example using a vehicle for travel's, product delivery, etc., there will be a separate premium and policies which is to be registered while taking the policy. However, nowadays some insurers are offering supplemental insurance products that extend coverage for vehicle owners by providing ride-sharing services.

2.4 Is auto insurance mandatory?

Auto insurance requirements vary from state to state. If we are taking the vehicle from some other financing providers, the lender may also have their requirements. In general, every state requires vehicle owners to carry: Bodily injury liability, Property damage liability, and medical or personal injury protection (PIP) in addition to these the typical coverages by auto insurer industries are collision, comprehensive, and glass coverage. There is one other special kind of insurance namely gap insurance.

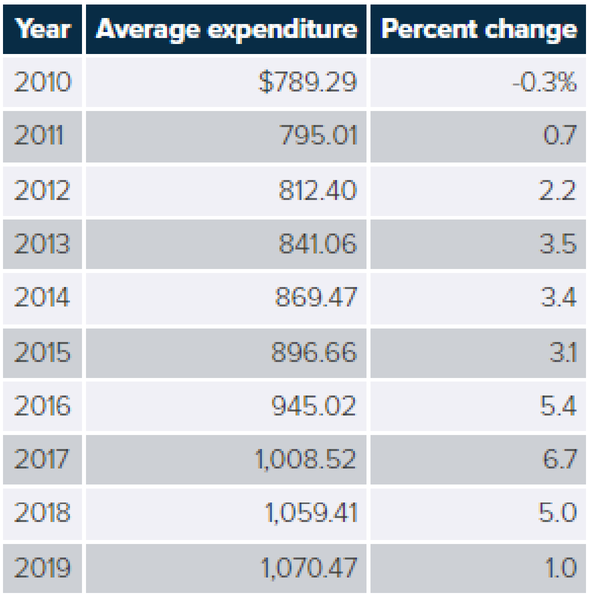
Gap Insurance: Collision and Comprehensive insurance will cover only the market value of the car but not what we paid for it. The depreciation value of new cars is high in case the vehicle is stolen or totalled in that case there may be a gap between the amount that we owe and insurance coverage. To fill this gap we might look into purchasing additional gap insurance to pay the difference. For leased vehicles generally, it will be rolled into the lease payments.

2.5 Statistics of auto insurance

Countrywide in USA, the automobile insurance expenditure rose from $1,059.41 in 2018 to $1,070.47 in 2019 which is 1.0 percent. This is the data according to the National Association of Insurance Commissioners. As per the data of the 2019 report, the average expenditure was highest in Louisiana, followed by Michigan and New York.

As per the study by 'AAA'S 2021 you're driving costs,' the average cost to own and operate a 2021 model vehicle was $9666 when the vehicle is driven 15000 miles per year. The average cost of insurance for a medium sedan model was $1400 for full coverage whereas for a medium SUV model car is $1300. As per the analysis of 2019 NAIC data, 79 percent of the insured drivers purchased comprehensive coverage in addition to liability insurance and 75 percent bought collision coverage.

The average expenditures for auto insurance from 2010 to 2019 are stated below in the table along with the percentage change. This is the data collected from the source 2022 National Association of Insurance Commissioners (NAIC).



Year	Average expenditure	Percent change
2010	$789.29	-0.3%
2011	795.01	0.7
2012	812.40	2.2
2013	841.06	3.5
2014	869.47	3.4
2015	896.66	3.1
2016	945.02	5.4
2017	1,008.52	6.7
2018	1,059.41	5.0
2019	1,070.47	1.0

Table 2.5.1 Average expenditures from 2010 to 2019

2.6 Understanding the features

Till now we understood the insurance industry and some other statistics etc. Now let us see some details about the features we had in our dataset. There are 40 columns in the dataset.

months_as_customer : Number of months they are a customer of the company.
Age : The age of the policy holder.
policy number : The policy number of the policy holder w.r.t company.
policy_bind_date : The moment when the coverage goes into force.
policy_state : The state of the insurance policy.
policy_csl : Provides the limit for coverage for all components of a claim.
policy_deductable : The amount to be spent for claiming insurance.
policy_annual_premium : The yearly premium amount to be paid for insuring.
umbrella_limit : The additional amount that can be claimed.
insured_zip : The zip code of location
insured_sex : Gender of the policyholder.
insured_education_level : The educational qualification of the policyholder.
insured_occupation : Occupation of the policyholder.
insured_hobbies : Hobbies of the policyholder
insured_relationship : Relationship status of policyholder
capital-gains : The profit arising on receipt of the claim.
capital-loss : The loss incurred when sold for less cost than purchased
incident_date : The date when the incident occurred.
incident_type : Number of vehicles involved.
collision_type : The type of collision.
incident_severity : The severity of the incident occurred.
authorities_contacted : The authorities that are contacted about the incident.
incident_state : The state where the incident occurred
incident_city : The city where the incident occurred.
incident_location : The exact address where the incident took place.
incident_hour_of_the_day : The time of the day when the incident happened.
Num_of_vehicles_involved: The exact number of vehicles involved in the incident.

property_damage	: There is damage to property or not.
bodily_injuries	: Number of injuries that occurred to the body.
Witnesses	: Number of witnesses, who have seen the occurrence of an incident
police_report_available	: Whether a police report is available or not.
total_claim_amount	: The total amount claimed for the incident.
injury_claim	: The amount claimed for injuries
property_claim	: The amount claimed for property
vehicle_claim	: The amount claimed for vehicle
auto_make	: Name of the vehicle manufacturer company
auto_model	: The model of the vehicle
auto_year	: The year of manufacturing of the vehicle.
fraud_reported	: Whether the claim made is fraud or not.
_c39	: Unknown (Even no data in the column)

So we research the above dataset to develop the Machine Learning model for intelligent prediction of the fraudulent insurance claim.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

As discussed above in 1.5, there are several phases in the methodology we carried out the research. We have also seen the literature review part, now we will discuss the detailed requirements, the procedure followed in building the model along with the observations and testing the models to find the best model.

Hardware and Software Requirements:

Hardware:

Processor – i5

Ram – 4GB

Softwares:

Anaconda framework – Jupyter Notebook,

Python,

GitHub account,

Packages:

Pandas,

Numpy,

Matplotlib,

Seaborn,

Scipy,

Sklearn,

3.2 Steps in data science

There are a lot of steps involved while solving any problem using data science methodology.

Data Science Process is an agile, iterative methodology to deliver predictive analytics solutions and intelligent applications efficiently. It has a well-structured process and architecture to be followed for better working and getting efficient results from the project. The different steps involved in the Data Science project are shown in below figure 3.1.1.

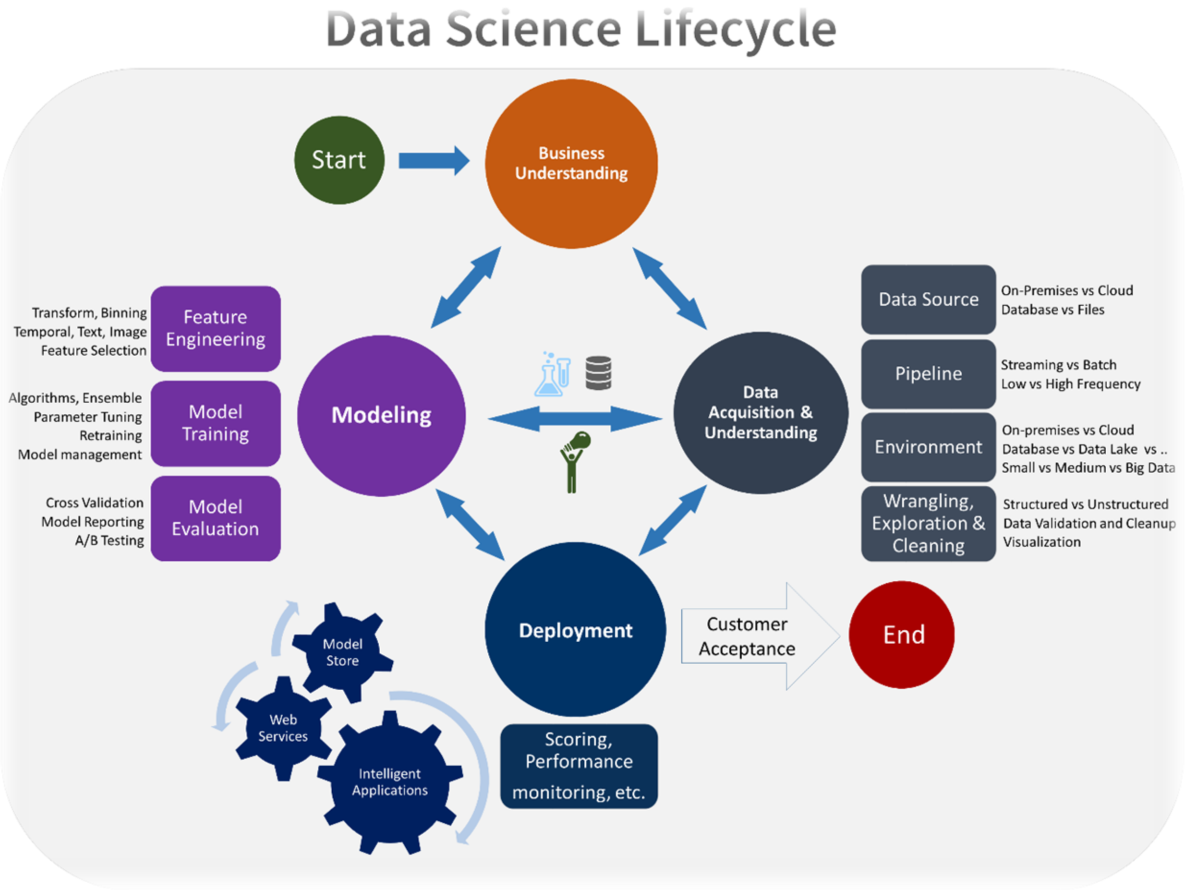


Fig. 3.2.1 Data Science Life Cycle

Let us discuss each step involved in the life cycle:

- Business Understanding: It is important to understand the business industry and what specific problem we are going to address and solve. Therefore we can collect the specific data required to solve that issue we are focusing to fix. We should be in a situation to interpret the result we got, and justify how it is accurate, efficient, and helps to solve the business problem.
- Data Mining: In this step based on our understanding of the business problem, we will collect the required data to analyze and fix the issue. There are various steps involved in this. They are as shown in fig. 3.1.2.

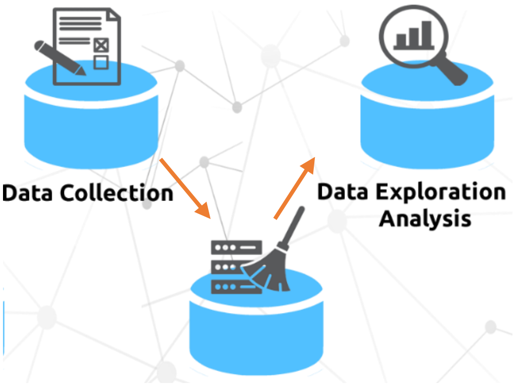
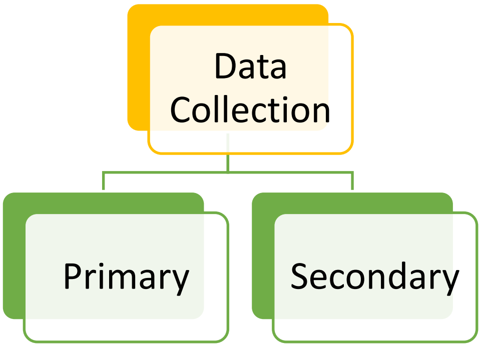


Fig. 3.1.2 Data Mining Steps

(a) Data Collecting: As per understanding of the business problem we will collect the required data. There are different methods we can use to collect the required data, which are generally classified into primary and secondary methods. The different techniques are collecting data from the web by performing web scraping, collecting from different existing sources, using the existing data of a company, or getting the data from the company for whom we are doing the project.



PRIMARY	SECONDARY
MORE TIME	LESS TIME COMPARATIVELY
Surveys, Polls, Experiments	Secondary Source, Readily available
Quantitative & Qualitative	No special methods

Table 3.1.a: Types of data collection

(b) Data Cleaning: After collecting the data we should understand the data precisely. The data collected might not be in a well-structured way always so we need to clean it first. Cleaning the data has many steps and there is no fixed methodology for it, e.g., changing the data type if it is wrongly interpreted by the system there may be several reasons causing that, removing the unnecessary columns, removing the duplicates, filling the missing values, etc. So it is important to understand the about business which helps us in this step which columns to be removed, how to deal with the missing values and other benefits.

(c) Data Exploration: Now we will analyse the cleaned data using various visualization and statistical techniques to observe the underlying trends, relations, and various aspects and get answers to many questions as per the business need and problems faced.

- Modelling:
  The next step after data mining is our modelling part, there are several steps involved here also like feature engineering, training, and testing the data.

(1) Feature Engineering: This is the step where we find the important features relating to the problem. We can find these features using several techniques

(2) Model training: Once the data is cleaned and found the important features then we will split the features and the target variable which is to be predicted. We will train the model with this data and predict the output. We will generally make different models using different algorithms.

(3) Model Testing: The developed models are then tested for their performance, i.e. how accurate they are in predicting the output. Then out of all the models developed we will select the best model which is predicting well compared to other models based on various factors as per business need. Then we will try to further improve the performance of the best model by hyper-parameter tuning.

- Deployment: Once the best model is developed it should be saved in .pkl format such that the .pkl file can be used for deployment on the web. Deployment is important for seeing the output, the client or whoever ever accessing should be able to get the output by just giving inputs without much bothering about the internal workings. Deployment can be done using various frameworks like a flask, and Django. Even there are a few new libraries in python that make it possible like Streamlit to avoid the Html and front-end part and can be hosted in any of the cloud providers like Azure, AWS, etc. such that it can be accessible on the web.

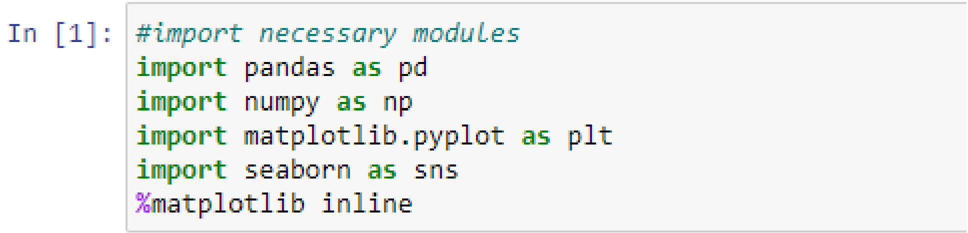
3.3 Data collection

In our case since it is a research purpose, we are using the existing data on the web. Resource of the data - https://github.com/sridharguvvala/Datasets/blob/main/8.Insurance.csv
The data we are using is from the US, which is a collection from various insurance companies collectively primarily made for research purposes.
The properties of the data we collected are:
Data Shape – 1000 x 40 (There are 1000 rows and 40 columns)
Data file format - .csv

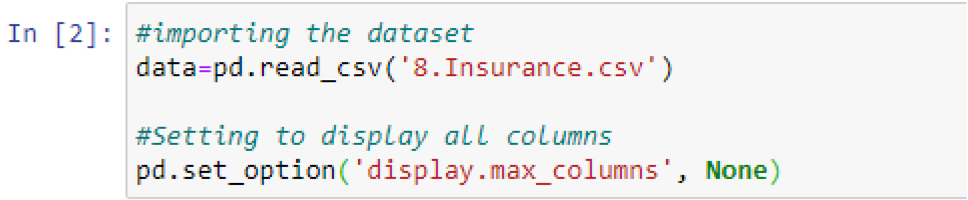
3.3.1 Importing and understanding the data

Now we had the data, let us understand it by importing the data file and observing them.

Importing the dataset in jupyter notebook and saving it to a variable



```
In [1]: #import necessary modules
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```



```
In [2]: #importing the dataset
        data=pd.read_csv('8.Insurance.csv')

        #Setting to display all columns
        pd.set_option('display.max_columns', None)
```

Fig. 3.3.1 Importing the required modules and reading the data

Cell [1]: We imported a few necessary modules/libraries for dealing with data and visualization

Pandas, Numpy – To deal with data and make changes in the data frame if needed.

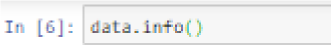
Matplotlib, Seaborn – These are the modules we use for creating graphs and visualizations to understand the insights of data.

Cell [2]: We read the data and stored it into a variable ‘data’ namely. Also, we had set an option to display all the columns.

8. Insurance.csv is the file name that is stored in the local machine at the current working directory

Column details:

Execute the below code below



```
In [6]: data.info()
```

By executing this code, we got the brief of the data columns like –

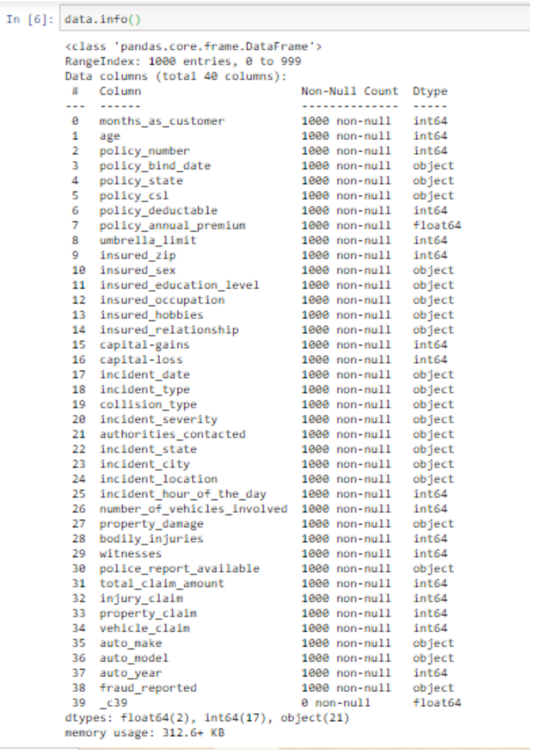
Range

Column name,

Number of non-null values and

Data type.

This is an important step to see whether there are any data type mismatches and to fix them. So we need to carefully observe the type of data present and verify it here.



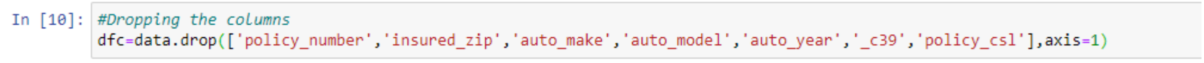
```
In [6]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 40 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   months_as_customer           1000 non-null   int64  
 1   age                          1000 non-null   int64  
 2   policy_number                1000 non-null   int64  
 3   policy_bind_date             1000 non-null   object 
 4   policy_state                 1000 non-null   object 
 5   policy_csl                   1000 non-null   object 
 6   policy_deductable            1000 non-null   int64  
 7   policy_annual_premium        1000 non-null   float64
 8   umbrella_limit               1000 non-null   int64  
 9   insured_zip                  1000 non-null   int64  
 10  insured_sex                  1000 non-null   object 
 11  insured_education_level      1000 non-null   object 
 12  insured_occupation           1000 non-null   object 
 13  insured_hobbies              1000 non-null   object 
 14  insured_relationship         1000 non-null   object 
 15  capital-gains                1000 non-null   int64  
 16  capital-loss                 1000 non-null   int64  
 17  incident_date                1000 non-null   object 
 18  incident_type                1000 non-null   object 
 19  collision_type               1000 non-null   object 
 20  incident_severity            1000 non-null   object 
 21  authorities_contacted        1000 non-null   object 
 22  incident_state               1000 non-null   object 
 23  incident_city                1000 non-null   object 
 24  incident_location            1000 non-null   object 
 25  incident_hour_of_the_day     1000 non-null   int64  
 26  number_of_vehicles_involved  1000 non-null   int64  
 27  property_damage              1000 non-null   object 
 28  bodily_injuries              1000 non-null   int64  
 29  witnesses                    1000 non-null   int64  
 30  police_report_available      1000 non-null   object 
 31  total_claim_amount           1000 non-null   int64  
 32  injury_claim                 1000 non-null   int64  
 33  property_claim               1000 non-null   int64  
 34  vehicle_claim                1000 non-null   int64  
 35  auto_make                    1000 non-null   object 
 36  auto_model                   1000 non-null   object 
 37  auto_year                    1000 non-null   int64  
 38  fraud_reported               1000 non-null   object 
 39  _c39                         0 non-null      float64
dtypes: float64(2), int64(17), object(21)
memory usage: 312.6+ KB
```

Observations:

- Our data is a mixture of integer, float, and object data types.
- Our target variable ‘fraud_reported’ is an object type. So we use the ‘Classification Model’.
- We see there are no null values in our brief data, but we found ‘?’ in some columns, so we need to replace them in the best possible way.
- Column policy_csl is shown as an object but we can see it contains numerical data. There might be chances of ‘?’ or any other special character or text present in that column. So it is shown as the object type. We need to find that and convert its type.
- Observe column 39: ‘_c39’- it has 0 non-null values, i.e. all the values are null. So we can drop that column straight away.
- **Assumptions:**
    - Assuming that, making fraud is a personal intention and that doesn’t depend on what kind of auto make, model and year. So I opt to drop these columns also.
    - Policy_number: It is a unique id given to the customer, to track the subscription status and other details of the customer. So we can drop this column, before dropping check if there are any duplicates in that column, if there are duplicates remove those rows by keeping one uniquely then drop the column.
    - Insured_zip: It is the zip code where the insurance was made. It doesn’t give any information or is useful to us. so we will drop this column too.
    - Policy_csl is Combined Single Limit, i.e. how much credible and what their premium covers, it is also not necessary. So we will drop it.

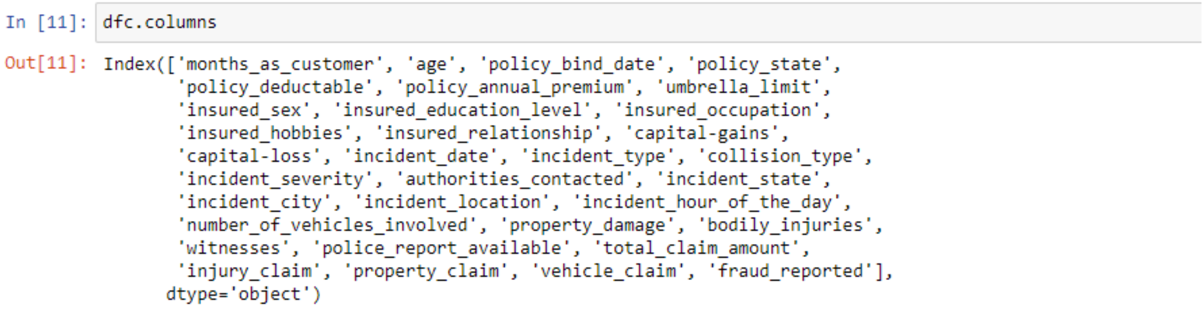
3.4 Data cleaning

Based on initial observation and understanding of data, we straight away removed a few columns which are not important for prediction.



```
In [10]: #Dropping the columns
         dfc=data.drop(['policy_number','insured_zip','auto_make','auto_model','auto_year','_c39','policy_csl'],axis=1)
```

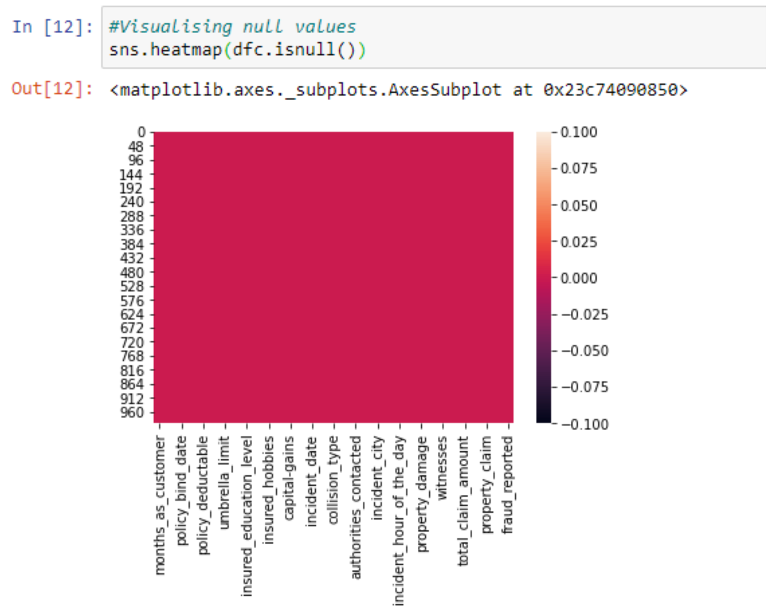
Drop is the command used to remove the columns from existing data, and we stored this data into a new variable namely ‘dfc’.



```
In [11]: dfc.columns

Out[11]: Index(['months_as_customer', 'age', 'policy_bind_date', 'policy_state',
                'policy_deductable', 'policy_annual_premium', 'umbrella_limit',
                'insured_sex', 'insured_education_level', 'insured_occupation',
                'insured_hobbies', 'insured_relationship', 'capital-gains',
                'capital-loss', 'incident_date', 'incident_type', 'collision_type',
                'incident_severity', 'authorities_contacted', 'incident_state',
                'incident_city', 'incident_location', 'incident_hour_of_the_day',
                'number_of_vehicles_involved', 'property_damage', 'bodily_injuries',
                'witnesses', 'police_report_available', 'total_claim_amount',
                'injury_claim', 'property_claim', 'vehicle_claim', 'fraud_reported'],
               dtype='object')
```

Listing the columns present in our new variable dfc to verify whether the columns are dropped or not.

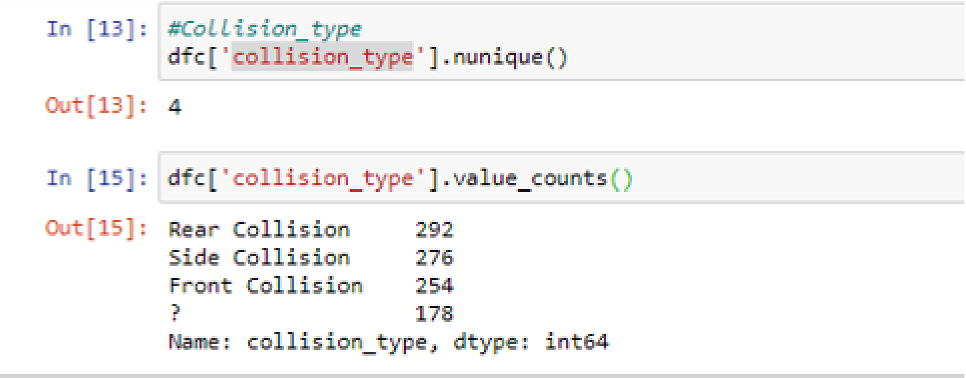
Checking for null values:



```
In [12]: #Visualising null values
         sns.heatmap(dfc.isnull())

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x23c74090850>
```

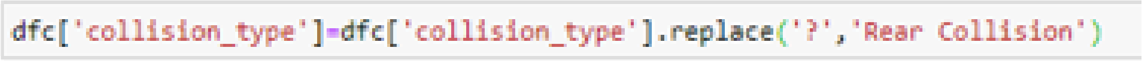
We can see there are no null values. But there are ‘?’, we are going to replace the ‘?’ with mean, median or mode, or any other based on the column.
We see ‘?’ present in many columns:
For E.g. we deal with one column collision type-
We can see there are 4 unique values in this column, as shown below. Front, Rear, Side, and ‘?’ Collisions. ‘?’ is there in 178 rows. This might not be filled or it is a manual error. Because the collision should be in one of the categories and must note while claimed.

The best, thumb-hand rule to deal with this kind in case of categorical values is to fill with the mode of the column.
So we will replace these ‘?’ with the most repeated type of collision type: Rear Collision.



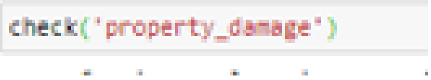
```
In [13]: #Collision_type
         dfc['collision_type'].nunique()

Out[13]: 4

In [15]: dfc['collision_type'].value_counts()

Out[15]: Rear Collision     292
         Side Collision     276
         Front Collision    254
         ?                  178
         Name: collision_type, dtype: int64
```

We use the code below to replace the ‘?’ symbol with the mode of that particular data in the given dataset.



```
dfc['collision_type']=dfc['collision_type'].replace('?','Rear Collision')
```

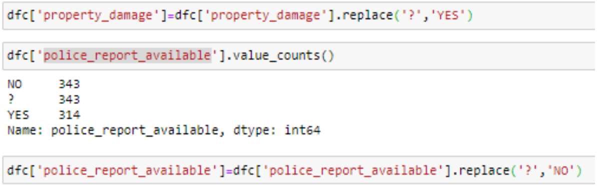
By executing the above code, the symbol ‘?’ is replaced with the text ‘Rare Collision’ and is saved into the same column ‘collision_type’.
Similarly, we do fill the ‘?’ with the mode of that column or by analyzing the column. I had filled them with mode.



```
check('property_damage')
```

Since there are many columns, to reduce the repetition of code always, I defined my function namely ‘check(feature_name)’. Basically what this code is it will check the data type of the column and if there is any ‘?’ in that column it will replace with mode if the feature is a categorical column and replace it with mean if it is not a categorical column.

In Property damage ‘?’ is replaced with YES. Since we observed that ‘?’ have more fraud claims. By comparing with other data, claims with yes are more fraud. So we replaced that with YES.



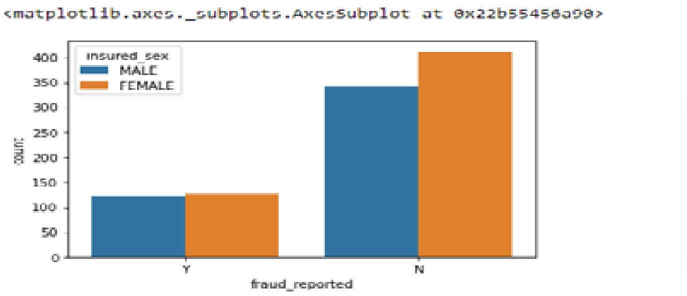
```
dfc['property_damage']=dfc['property_damage'].replace('?','YES')

dfc['police_report_available'].value_counts()

NO     343
?      343
YES    314
Name: police_report_available, dtype: int64

dfc['police_report_available']=dfc['police_report_available'].replace('?','NO')
```

Similarly, the police report is also filled with NO, by using the above code. The same method is applied to the remaining columns to check and replace values. Now the data is cleaned, let us explore the data visually for more clear insights.

3.5 Data exploration

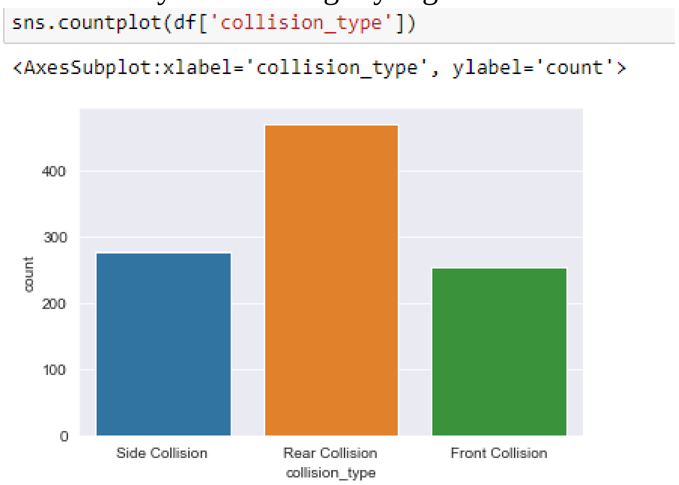
Now we will explore the data and try to understand some insights based on the visualizations.

Checking how many frauds occurred in the data we took for study:



```
<matplotlib.axes._subplots.AxesSubplot at 0x22b55456a90>
```

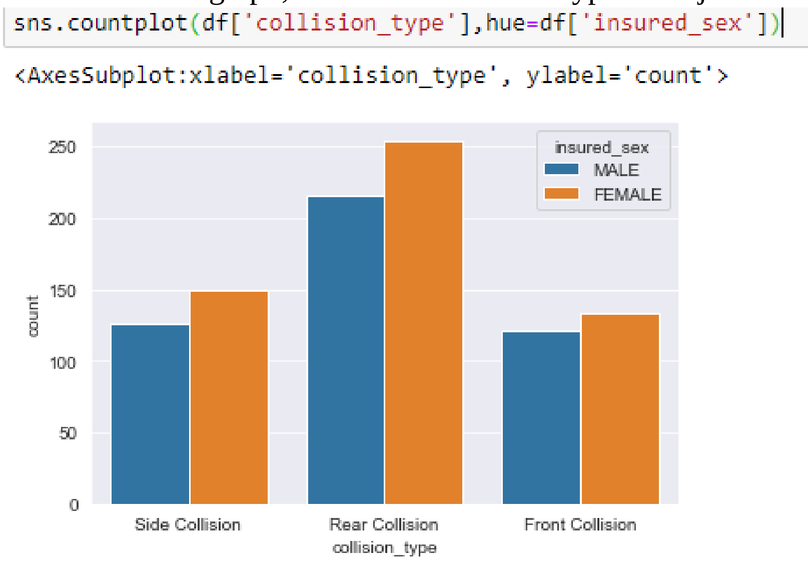
From the above graph, we can understand our data is unbalanced, i.e. the data consists more with no frauds and less with frauds occurred. Also, we can see that the percentage of fraud committed by males is slightly higher than females who committed fraud.



```
sns.countplot(df['collision_type'])
```

```
<AxesSubplot:xlabel='collision_type', ylabel='count'>
```

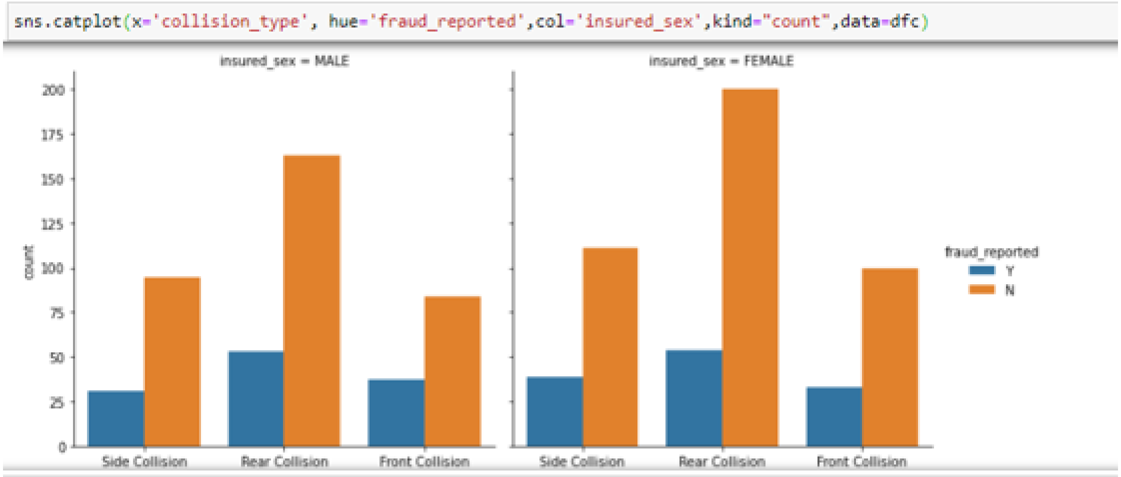
From the above graph, we can see collision type in major claims is Rear Collison.



```
sns.countplot(df['collision_type'],hue=df['insured_sex'])
```

```
<AxesSubplot:xlabel='collision_type', ylabel='count'>
```

We can see females are making more accidents than males, irrespective of collision type.



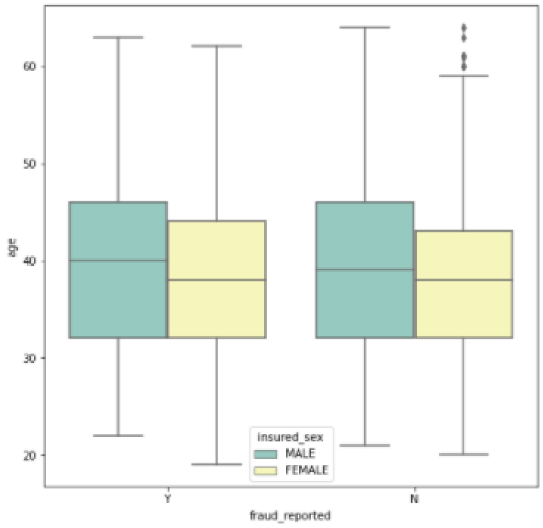
```
sns.catplot(x='collision_type', hue='fraud_reported',col='insured_sex',kind="count",data=dfc)
```

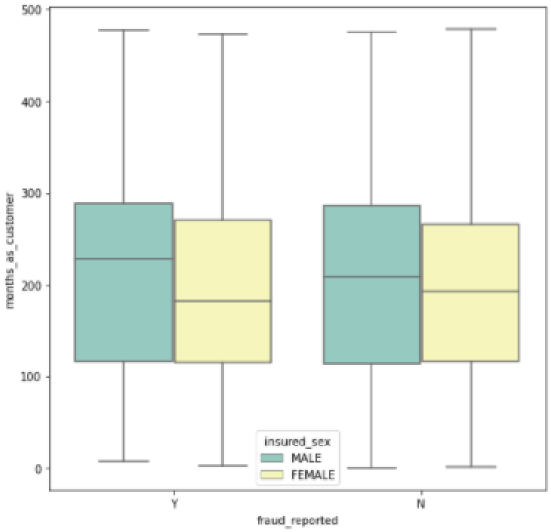
From the above graph, we can say that claims made as rare collision types either by males or females have more percent chance of being fraudulent compared to other types of collision types reported. Also, the majority of collisions occurred are Rear collisions.

Average age and months being a customer: 38.9 yrs. old, 203.95 months as a customer

```
Brief of age: count    1000.000000
mean        38.948000
std          9.140287
min         19.000000
25%         32.000000
50%         38.000000
75%         44.000000
max         64.000000
Name: age, dtype: float64
skewness: 0.47898804709224163
unique values count in age: 46
```

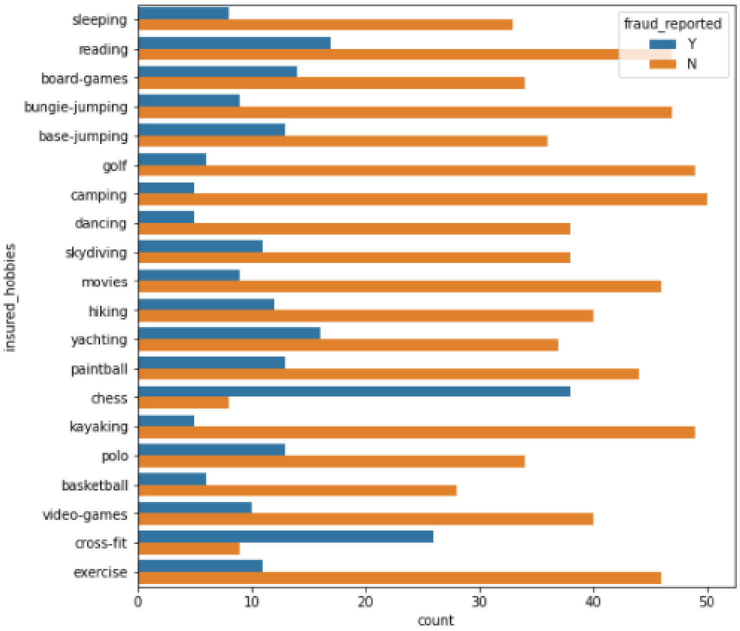
```
Brief of months_as_customer: count    1000.000000
mean       203.954000
std        115.113174
min          0.000000
25%        115.750000
50%        199.500000
75%        276.250000
max        479.000000
Name: months_as_customer, dtype: float64
skewness: 0.3621768477780205
unique values count in months_as_customer: 391
```



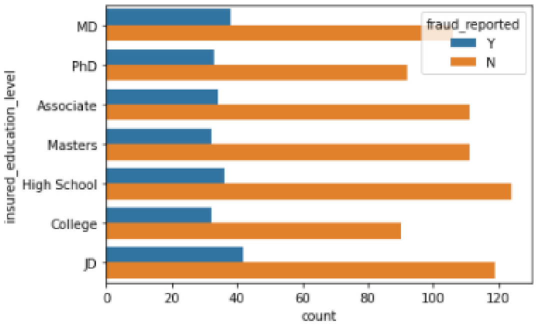


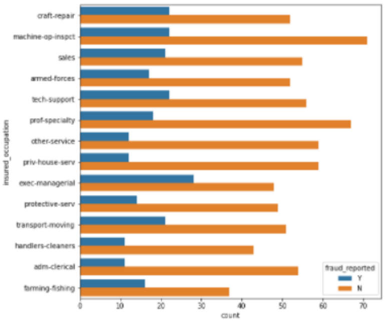
We can see males had done frauds from the age of 22/23 approx., females had done from teens, approx. from 18 yrs. Frauds committed by clients from their 1st month onwards.

Hobbies: People with a hobby of chess and cross-fit had done more fraud claims.



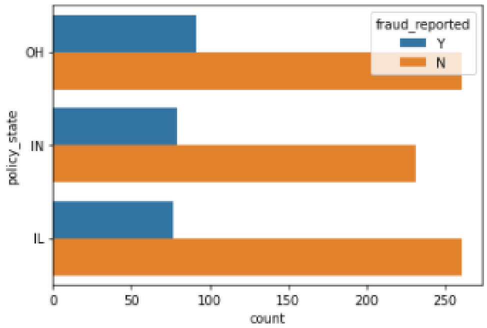
Education level: We can see, that those who studied JD, and MD has done more frauds comparatively.

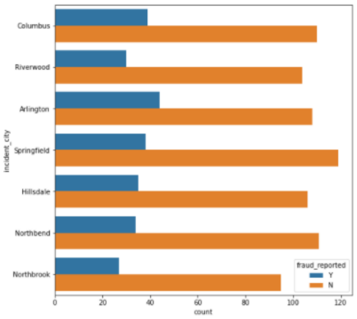




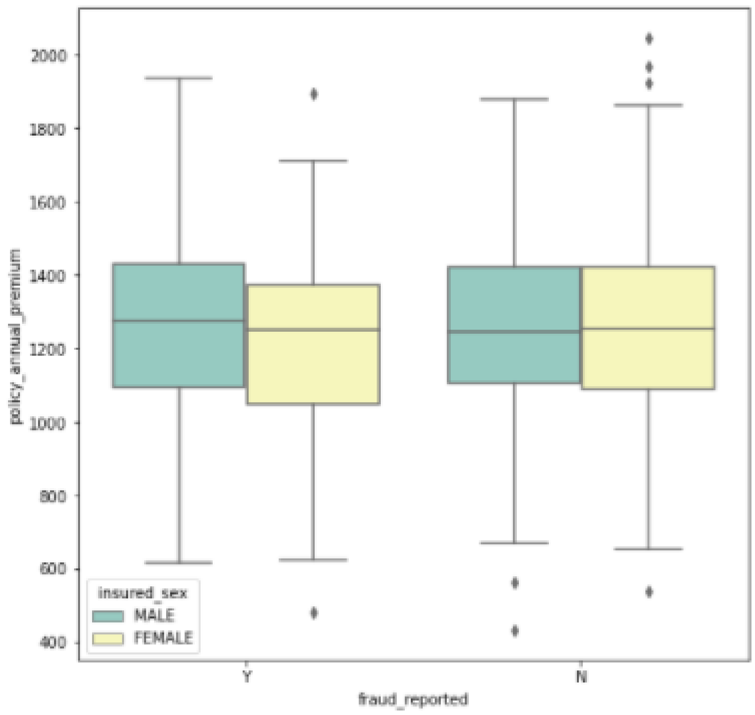
Occupation: People of job exec. manager are doing more frauds.

More frauds happened in the state of OH and cities: Arlington, Columbus

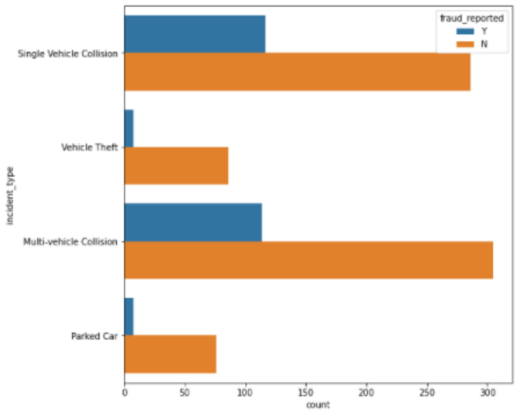


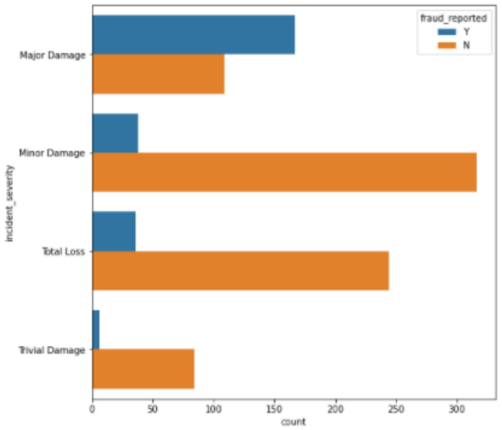


Policy premium: We have seen that people with a premium of above-average are committing fraud

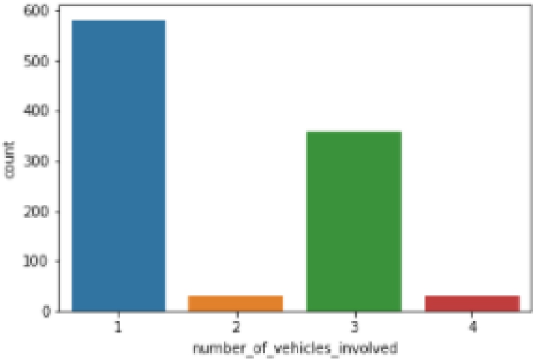


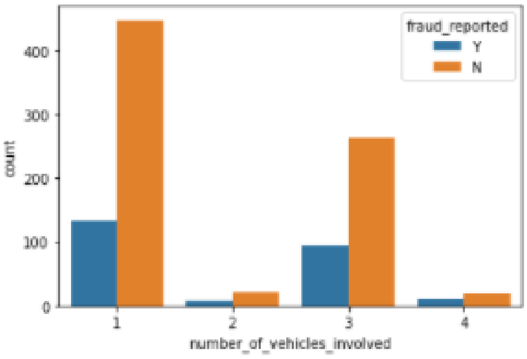
People who claimed incident types such as Single-vehicle collisions or multiple vehicles and major damage collisions are more fraudulent claims.



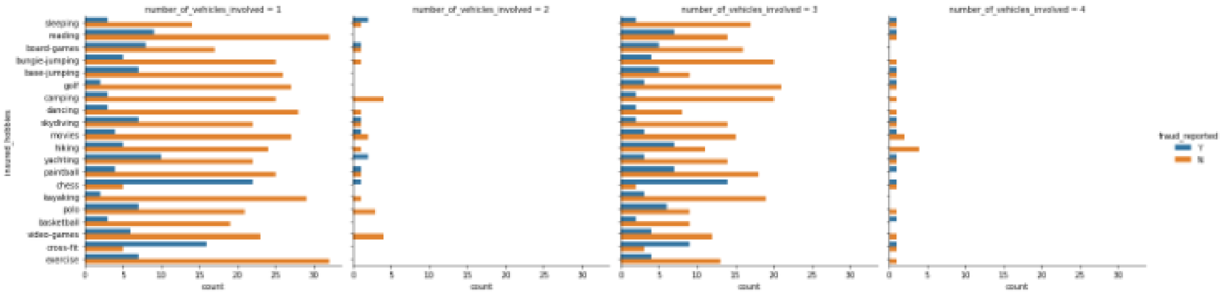


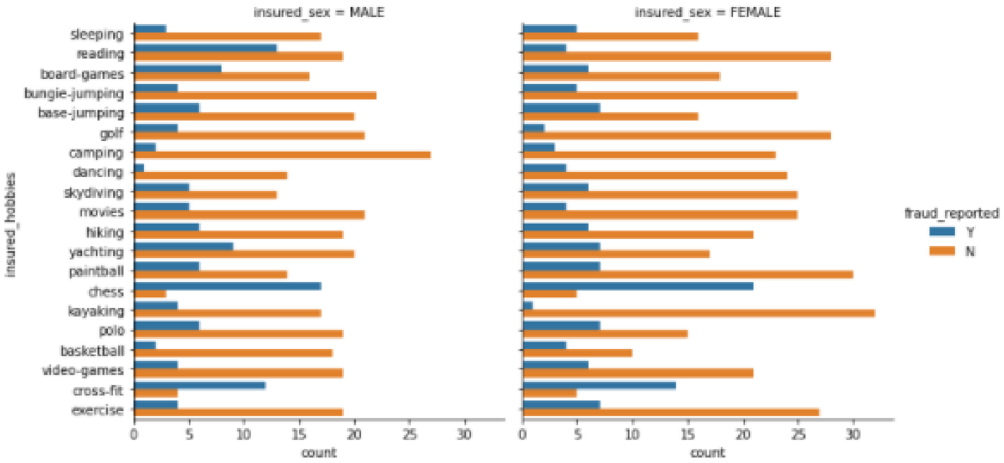
The majority of the incidents are 1 and 3 vehicles, and also 1 involved are more fraud claims, but if we see carefully even though 2 and 4 vehicles involved claims are less, 50% of them are fraud.

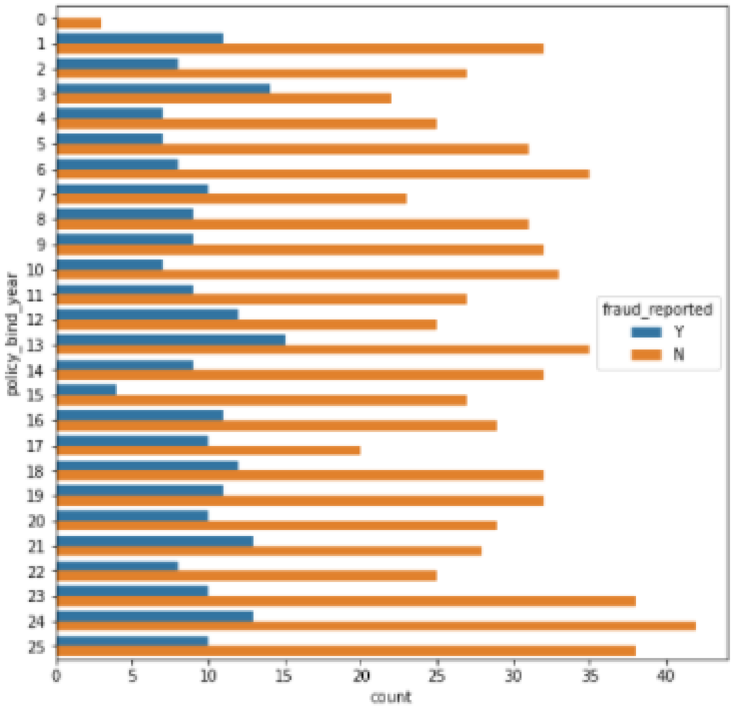




Observing the fraud basis on no. of vehicles and Hobbies, Gender and Hobbies.







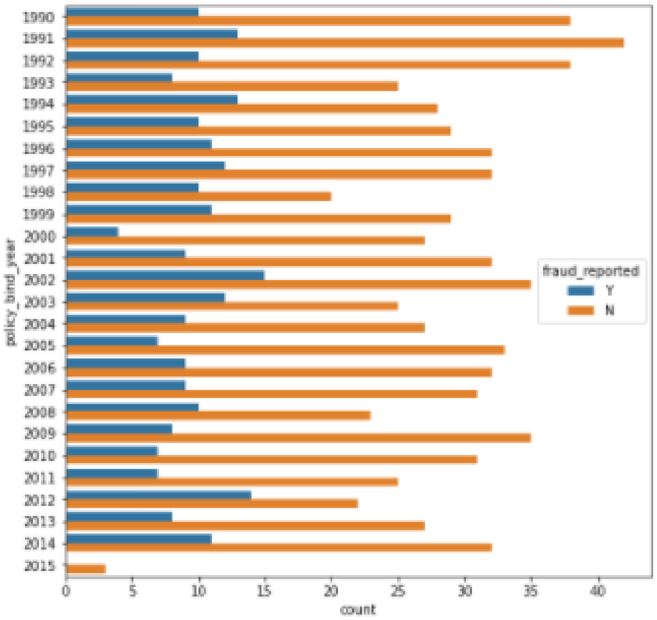
Based on number of years of policy binded

Similarly, we observe and get insights from other columns.

We removed two more columns, incident date and incident location, which are not necessary and have almost unique values in each row.

We made a new column policy bind year from the policy bind date and dropped the original column.

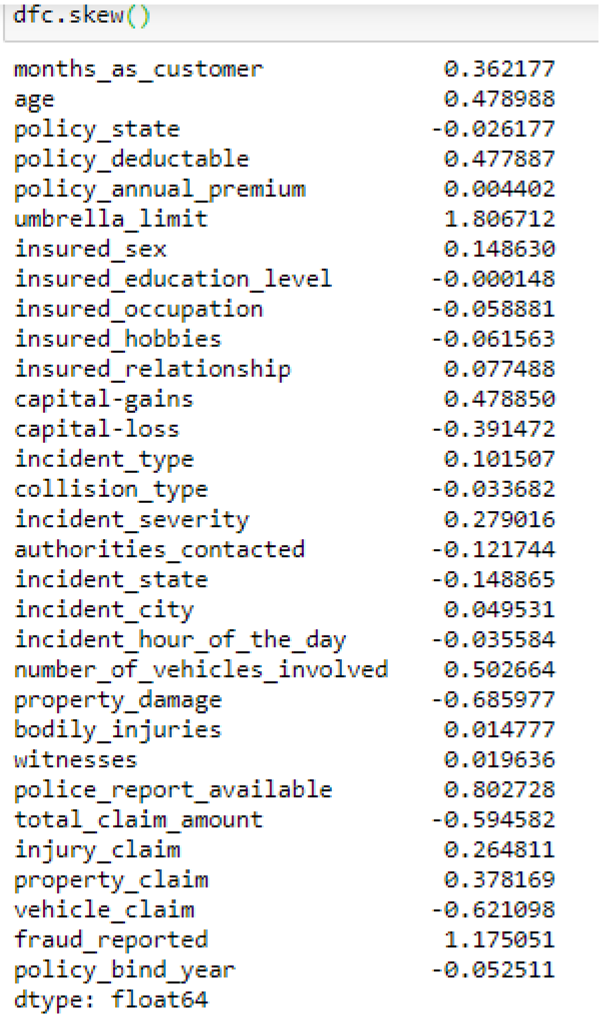
From that, we observed policies were made from the years 1990 to 2015, and more frauds are of those who made insurance in the year 1991, 2002, and 2012.



So to make it simple, we found the difference of number of years from 2015.

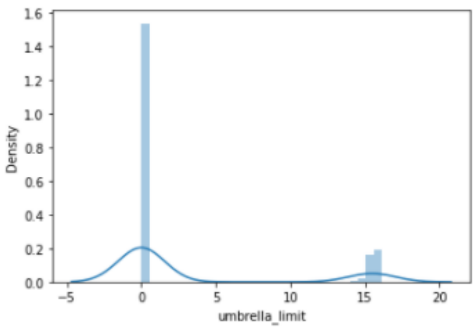
We made all the columns encoded and find the correlation.

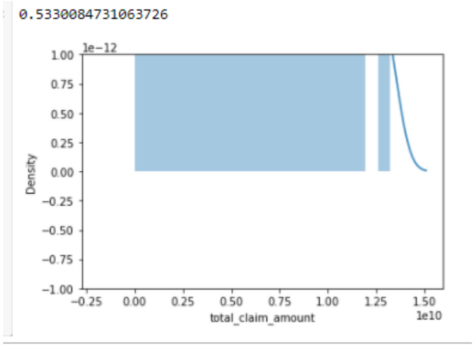
Checking the skewness and reducing skewness in columns umbrella limit, total claim amount



```
dfc.skew()
```

```
months_as_customer             0.362177
age                            0.478988
policy_state                  -0.026177
policy_deductable              0.477887
policy_annual_premium          0.004402
umbrella_limit                 1.806712
insured_sex                    0.148630
insured_education_level       -0.000148
insured_occupation            -0.058881
insured_hobbies               -0.061563
insured_relationship           0.077488
capital-gains                  0.478850
capital-loss                  -0.391472
incident_type                  0.101507
collision_type                -0.033682
incident_severity              0.279016
authorities_contacted         -0.121744
incident_state                -0.148865
incident_city                  0.049531
incident_hour_of_the_day      -0.035584
number_of_vehicles_involved    0.502664
property_damage               -0.685977
bodily_injuries                0.014777
witnesses                      0.019636
police_report_available        0.802728
total_claim_amount            -0.594582
injury_claim                   0.264811
property_claim                 0.378169
vehicle_claim                 -0.621098
fraud_reported                 1.175051
policy_bind_year              -0.052511
dtype: float64
```

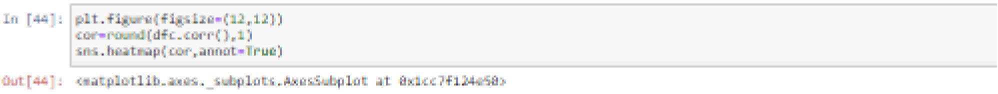




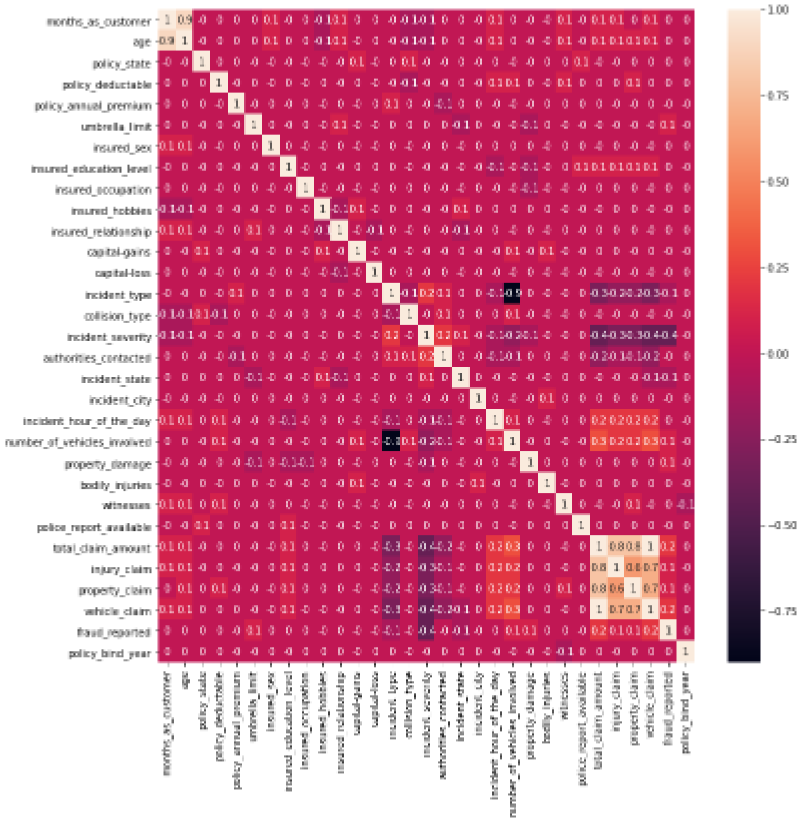
```
0.5330084731063726
```

Skewness reduced to 1.4 from 1.8, Skewness reduced to 0.5 from -0.59

Plot of correlation:



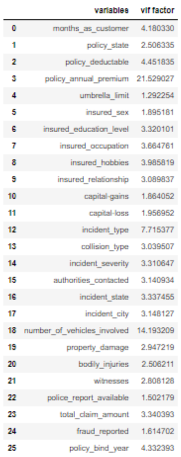
```
In [44]: plt.figure(figsize=(12,12))
         cor=round(dfc.corr(),1)
         sns.heatmap(cor,annot=True)

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x1cc7f124e58>
```



Observations:

- ➢ Age & months as a customer are highly correlated, we will drop the column age.
- ➢ Total claim amount, injury claim, a property claim, and vehicle claim are correlated to each other. So we will drop all those columns except the Total claim amount
- ➢ Incident hour of the day doesn't have any correlation with fraud. So we will drop it.\
- ➢ Many columns have no correlation with fraud reported

Let us drop these columns, find the VIF to know the multicollinearity, and drop other columns.

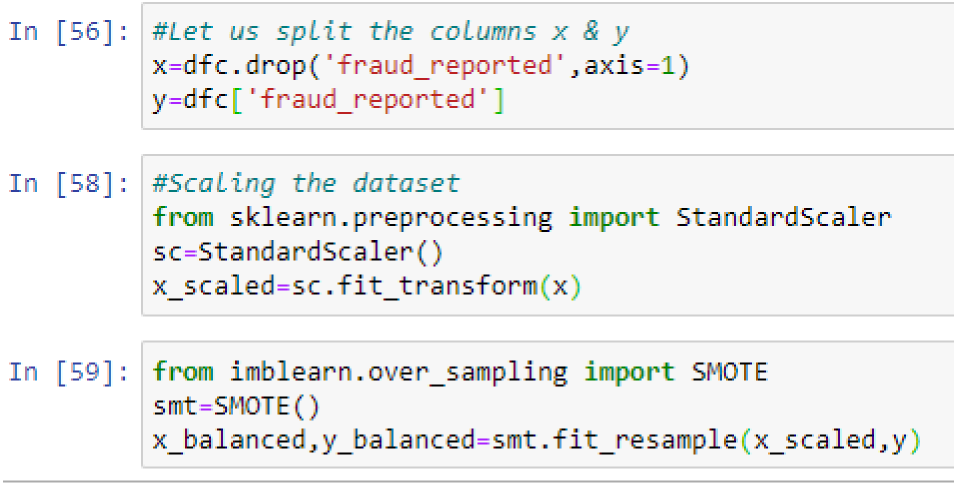


	variables	vif factor
0	months_as_customer	4.180330
1	policy_state	2.506335
2	policy_deductable	4.451835
3	policy_annual_premium	21.529027
4	umbrella_limit	1.292254
5	insured_sex	1.895181
6	insured_education_level	3.320101
7	insured_occupation	3.664761
8	insured_hobbies	3.985819
9	insured_relationship	3.089837
10	capital-gains	1.864052
11	capital-loss	1.956952
12	incident_type	7.715377
13	collision_type	3.039507
14	incident_severity	3.310647
15	authorities_contacted	3.140934
16	incident_state	3.337455
17	incident_city	3.148127
18	number_of_vehicles_involved	14.193209
19	property_damage	2.947219
20	bodily_injuries	2.506211
21	witnesses	2.808128
22	police_report_available	1.502179
23	total_claim_amount	3.340393
24	fraud_reported	1.614702
25	policy_bind_year	4.332393

By viewing VIF I dropped columns

Number of vehicles involved, premium deductables, capital gain and capital loss.

Finally, two last steps before making the model, Split x & y, then scale the x, and finally balance the dataset.



```
In [56]: #Let us split the columns x & y
         x=dfc.drop('fraud_reported',axis=1)
         y=dfc['fraud_reported']

In [58]: #Scaling the dataset
         from sklearn.preprocessing import StandardScaler
         sc=StandardScaler()
         x_scaled=sc.fit_transform(x)

In [59]: from imblearn.over_sampling import SMOTE
         smt=SMOTE()
         x_balanced,y_balanced=smt.fit_resample(x_scaled,y)
```

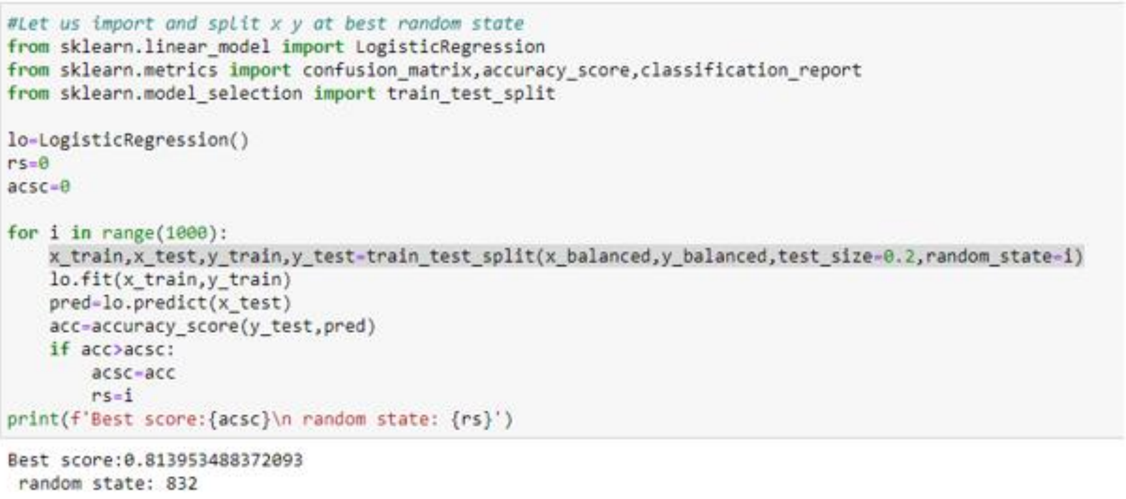
X features, y is our target.

Brief of observations and steps – In this section of Model visualization, we visualized the different columns, and trends got insights from underlying data, how they are related, and found the few features which are important to determine the claim, also found the columns with high multicollinearity and removed those features. To find this we made the categorical columns encoded. We even found the skewness and removed it in the columns wherever necessary. Then finally we divided the target column from features and then scaled the features and then balanced the data to avoid the problems that will occur due to unbalanced data.

3.6 Model building

Now we have the data cleaned, only with required columns, encoded the categorical features, removed outliers, reduced skewness if any, and scaled and balanced dataset such that we can use this neat and well-prepared data for model building.

As discussed the first step in model building is training the data. Initially, we need to split the data with us into train and test. We will import the required modules to perform this action. ‘train_test_split’ from ‘sklearn.model_selection’ is the required one.



```
#Let us import and split x y at best random state
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
from sklearn.model_selection import train_test_split

lo=LogisticRegression()
rs=0
acsc=0

for i in range(1000):
    x_train,x_test,y_train,y_test=train_test_split(x_balanced,y_balanced,test_size=0.2,random_state=i)
    lo.fit(x_train,y_train)
    pred=lo.predict(x_test)
    acc=accuracy_score(y_test,pred)
    if acc>acsc:
        acsc=acc
        rs=i
print(f'Best score:{acsc}\n random state: {rs}')
```

```
Best score:0.813953488372093
 random state: 832
```

Let us understand the above block of code.

Since our model output is like yes or no it will fall under the classification model. So we imported the “Logistic Regression’ to predict.

We imported some other metrics that are required to evaluate or test the model.

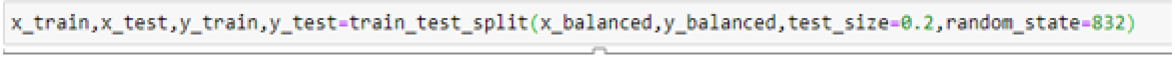
To split the data into train and test we imported train_test_split. In general, we need to split the data in such a way majority of data should be given for the model to learn, and the remaining data to be used for testing the performance of the model.

General ratio: 75 – 80 % of data for training and the remaining 20 – 25% data for testing.

In the above, we were given 80% data for training, since we have fewer data.

A random state is used for shuffling data. I implemented the above code to find the best possible shuffling so that the logistic regression model could perform well.

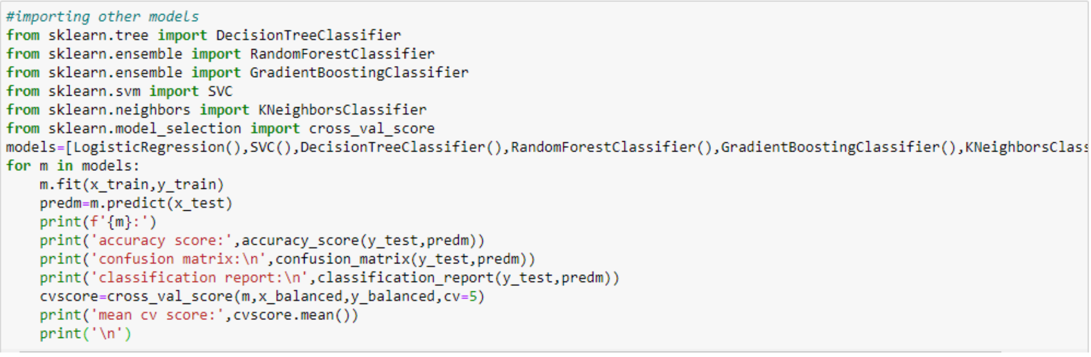
We can see the output of the logistic regression is performing well with its maximum best accuracy of 81.3% at the random state 832.



```
x_train,x_test,y_train,y_test=train_test_split(x_balanced,y_balanced,test_size=0.2,random_state=832)
```

So we split the data train and test as shown above at random state 832.

Now we try to build some other models to predict.



```
#importing other models
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
models=[LogisticRegression(),SVC(),DecisionTreeClassifier(),RandomForestClassifier(),GradientBoostingClassifier(),KNeighborsClass
for m in models:
    m.fit(x_train,y_train)
    predm=m.predict(x_test)
    print(f'{m}:')
    print('accuracy score:',accuracy_score(y_test,predm))
    print('confusion matrix:\n',confusion_matrix(y_test,predm))
    print('classification report:\n',classification_report(y_test,predm))
    cvscore=cross_val_score(m,x_balanced,y_balanced,cv=5)
    print('mean cv score:',cvscore.mean())
    print('\n')
```

Imported and built various models to predict the output as shown above.

The different models we built are Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, SVC, and KNN.

3.7 Model testing

Now the models built should be tested. There are different metrics to be compared and understood.

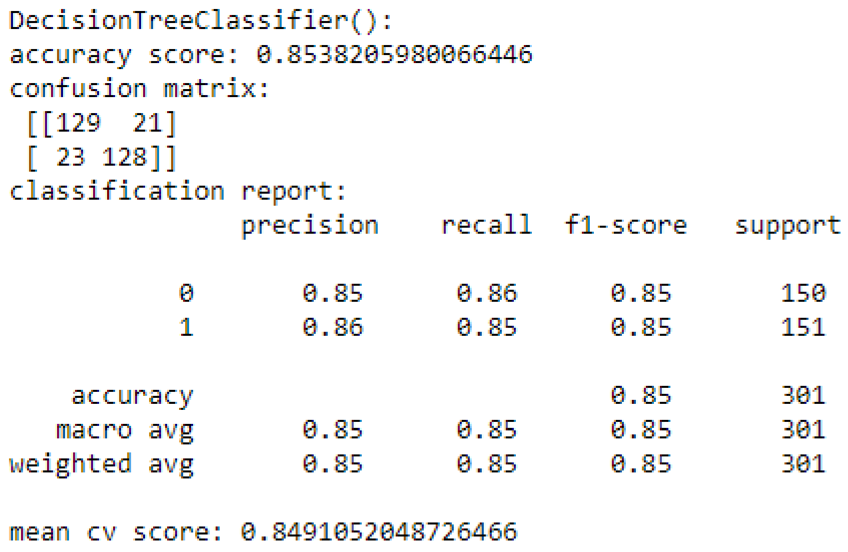
The different metrics are confusion matrix, f1 score, and accuracy report. Based on the industry and problem we are trying to fix, metrics and their output have weightage.

From the above code in the model building, we got results of accuracy score, f1 score, evaluation report, and cross-validation score of different models –

Model	F1 score	Cross-validation	Difference
Logistic Regression	81	74.5	6.5
Decision Tree Classifier	85	84.9	0.1
Random Forest Classifier	91	87	4
Gradient Boosting Classifier	91	88	3
K Neighbors Classifier	72	73	1
SVC	89	84.5	4.5

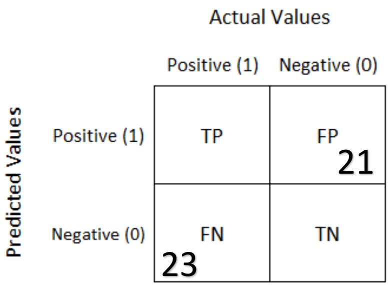
Table 3.7.1 Comparing different model's performance

From the above, we can see, that the Decision Tree is the best model with the least difference between CV score and F1 score. These are the metrics generated:



```
DecisionTreeClassifier():
accuracy score: 0.8538205980066446
confusion matrix:
 [[129  21]
 [ 23 128]]
classification report:
               precision    recall  f1-score   support

           0       0.85      0.86      0.85       150
           1       0.86      0.85      0.85       151

    accuracy                           0.85       301
   macro avg       0.85      0.85      0.85       301
weighted avg       0.85      0.85      0.85       301

mean cv score: 0.8491052048726466
```

From the confusion matrix, we can see



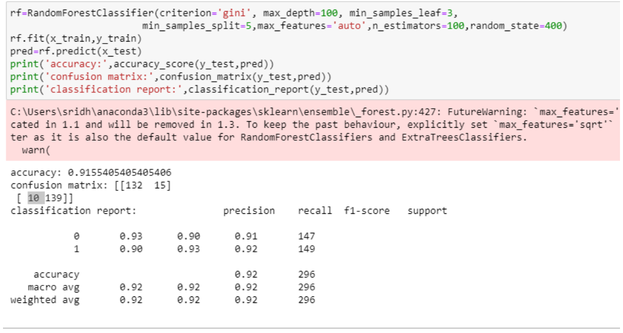
Our model is saying 23 members as not fraud where they are actually frauds. We need to reduce this for better results.

FP is ok because checking whether fraud or not is less cost to company than paying huge amount to the original fraud claim.

In our case we need least False Negative.

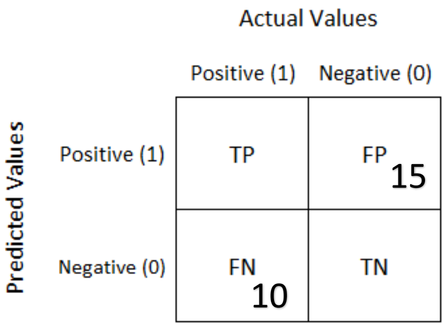
But wait there is almost a 6% difference in Decision Tree classifier, Random Forest, or Gradient Boosting. Let us observe their metrics once

If we see other models: It has a 3% difference but gives the least FN

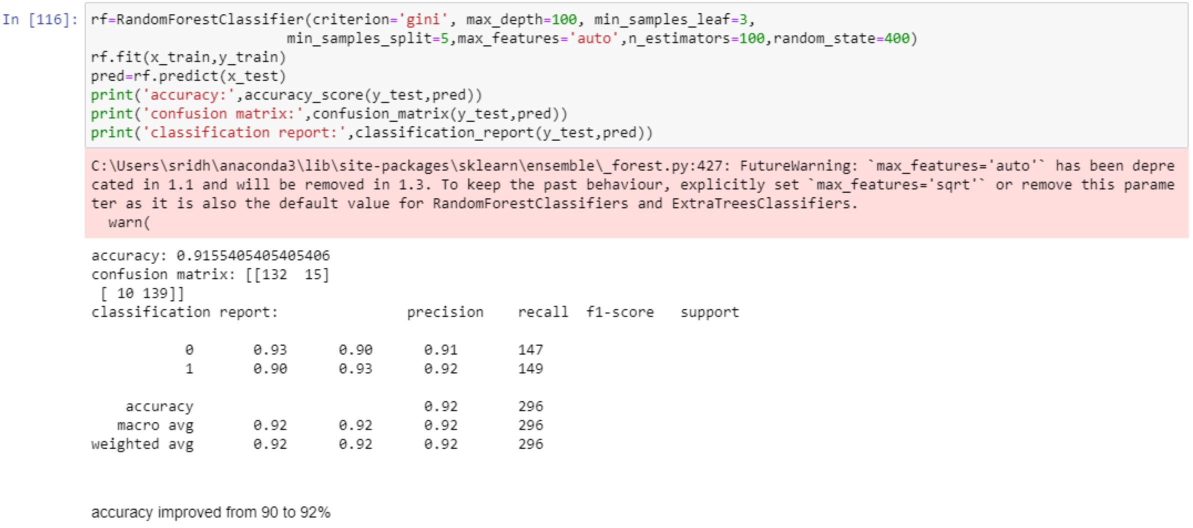


```
rf=RandomForestClassifier(criterion='gini', max_depth=100, min_samples_leaf=3,
                          min_samples_split=5,max_features='auto',n_estimators=100,random_state=400)
rf.fit(x_train,y_train)
pred=rf.predict(x_test)
print('accuracy:',accuracy_score(y_test,pred))
print('confusion matrix:',confusion_matrix(y_test,pred))
print('classification report:',classification_report(y_test,pred))
```

```
C:\Users\sridh\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:427: FutureWarning: `max_features='
cated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`
ter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(

accuracy: 0.9155405405405406
confusion matrix: [[132  15]
 [ 10 139]]
classification report:               precision    recall  f1-score   support

           0       0.93      0.90      0.91       147
           1       0.90      0.93      0.92       149

    accuracy                           0.92       296
   macro avg       0.92      0.92      0.92       296
weighted avg       0.92      0.92      0.92       296
```



By tuning the random forest model classifier: The model accuracy increased from 90 to 92% only. With FN as 10.

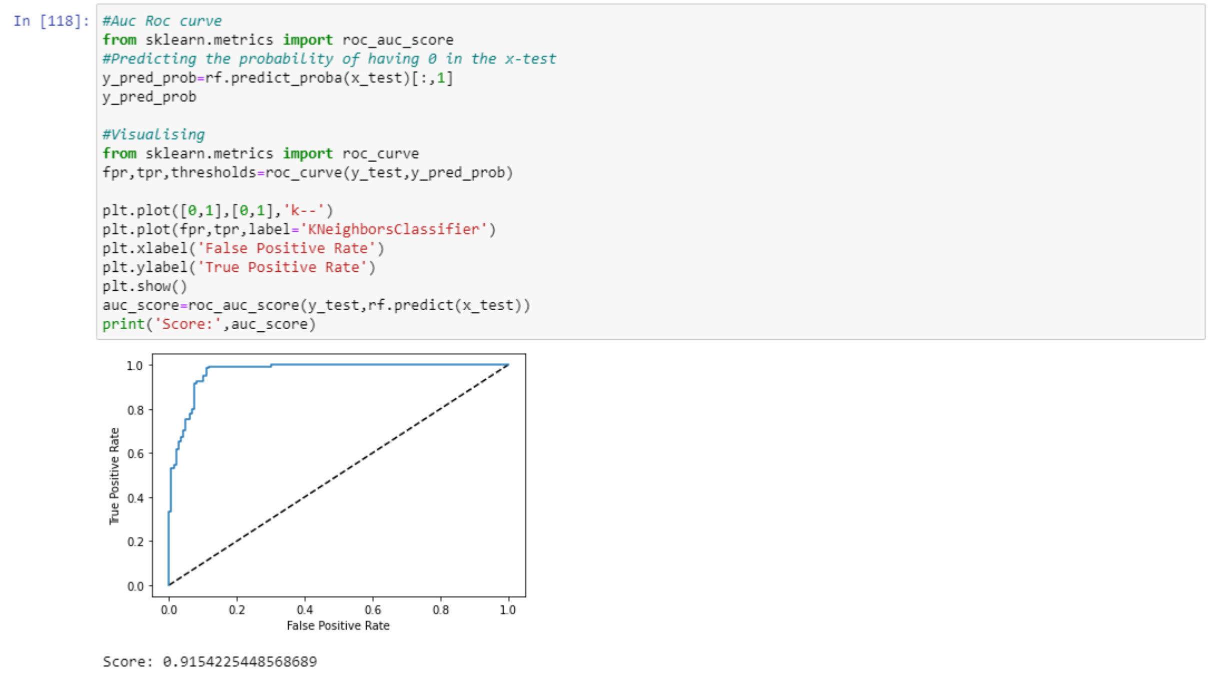


```
In [116]: rf=RandomForestClassifier(criterion='gini', max_depth=100, min_samples_leaf=3,
                                    min_samples_split=5,max_features='auto',n_estimators=100,random_state=400)
          rf.fit(x_train,y_train)
          pred=rf.predict(x_test)
          print('accuracy:',accuracy_score(y_test,pred))
          print('confusion matrix:',confusion_matrix(y_test,pred))
          print('classification report:',classification_report(y_test,pred))
```

```
C:\Users\sridh\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py:427: FutureWarning: `max_features='auto'` has been depre
cated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this parame
ter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(
```

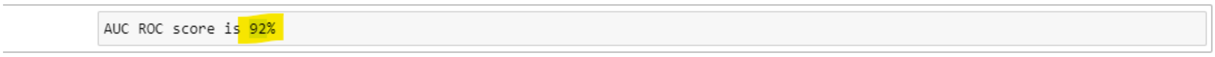
```
accuracy: 0.9155405405405406
confusion matrix: [[132  15]
 [ 10 139]]
classification report:               precision    recall  f1-score   support

           0       0.93      0.90      0.91       147
           1       0.90      0.93      0.92       149

    accuracy                           0.92       296
   macro avg       0.92      0.92      0.92       296
weighted avg       0.92      0.92      0.92       296
```

accuracy improved from 90 to 92%

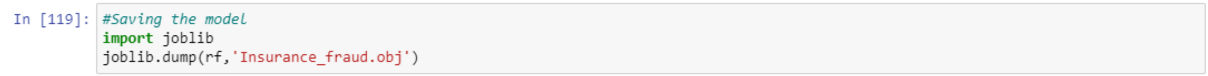


```
In [118]: #Auc Roc curve
          from sklearn.metrics import roc_auc_score
          #Predicting the probability of having 0 in the x-test
          y_pred_prob=rf.predict_proba(x_test)[:,1]
          y_pred_prob

          #Visualising
          from sklearn.metrics import roc_curve
          fpr,tpr,thresholds=roc_curve(y_test,y_pred_prob)

          plt.plot([0,1],[0,1],'k--')
          plt.plot(fpr,tpr,label='KNeighborsClassifier')
          plt.xlabel('False Positive Rate')
          plt.ylabel('True Positive Rate')
          plt.show()
          auc_score=roc_auc_score(y_test,rf.predict(x_test))
          print('Score:',auc_score)
```

```
Score: 0.9154225448568689
```



```
AUC ROC score is 92%
```



```
In [119]: #Saving the model
          import joblib
          joblib.dump(rf,'Insurance_fraud.obj')
```

Finally, we saved the best model i.e. Decision Tree Classifier as a .pkl file. So it can be given used for deployment purposes.

3.8 Summary

Finally, as per our main objective and the data science project life cycle steps, we understood the business problem, collected the data cleaned, observed the underlying trends, and built the machine learning models with various algorithms. Their performance is compared, the best model is selected and still, we tuned the model to improve its accuracy. The model we made gives the output with an accuracy of 92%.

REFERENCES

Wipro: Comparitive analysis of Machine Learning techniques for fraud detection.

IceAsher Chew: For real? Auto insurance fraud claim detection with Machine Learning

Plug & Play: Detecting insurance fraud with Machine Learning

Wallet hub: Types of car insurance

Payoda: Role of AI and ML in insurance fraud detection

Insurance Information Institute: Auto Insurance