**ANALYSIS OF CAR INSURANCE CLAIM PREDICTION**

Group project by:-

Ajay Sai Rudravaram,

Kiran Balija

Satya Durga prasad Vulli

Department: -

MSBA

Institution Name: -

Sacred Heart University

Course Number: -

BUAN-651 C (Intro to data and programming)

Professor’s Name: -

Max Yun

**RESEARCH QUESTIONS**

The questions we want to answer with our data are, which area cluster has more insurance claims? Which people of age driving the cars are claiming the insurance? What model and make of cars do the people in the cluster use? Owners of which car segment are claiming the insurance the most? Owners of cars with what type of features are claiming insurance? Cars with what safety rating in an area of the cluster has a high number of claims? How many non-corelative features of the dataset factor in for the outcome of the prediction?

**Importance**

These questions are of interest and are essential to answer because by gaining the information about which age of people are claiming the insurance, we can get an estimate on people of a particular range of age who will likely claim insurance. By knowing the model and make of the car people use in the cluster, we can get an estimate on which particular make and model of cars are insured highly. By obtaining information about the owners of various segments of cars claiming insurance, we can estimate which segments of cars are highly insured. By acquiring the information on cars with different features being insured, we can get an estimate on cars that have specific features or lack certain features that will likely get insured. By securing the information about the cars with different safety ratings in an area of the cluster, we can get an estimate on cars with what particular safety rating will likely get insured. By gathering information about which cluster area has more insurance claims, we can estimate which areas with similar conditions will have high insurance claims. The answers we gain to all these questions will determine whether the policyholder files a claim in the next six months.

**Hypotheses**

We expect to understand whether a policyholder will file a claim in the next six months by analysing the data, which includes several instances and features based on our research questions. The answers we attain will factor into the process of the prediction. This prediction will help the company to increase the claim process's efficiency.

**MOTIVATION**

The motivation behind the research questions was I used to commute to college on public transportation. Most of the time, due to several external factors, there was a delay in the transit, and I used to run late for my classes. Hence, to eradicate this occasional delay and dependency on public transportation, I decided to buy a car to commute to the college, depending on my schedule. However, buying a car is just one side of the coin. We need to pay for its insurance too. Most of the time, the premium we pay is very high and depends on several factors.

Similarly, the claiming of that insurance also depends on several factors. I wanted to know and understand what the factors were and how one can use these factors in predicting whether the policyholder makes a claim in the next six months. These factors help increase the efficiency of claim process.

**Background**

Extensive research has been done during the development of the research questions. In order to determine what could be the factors that could be used in prediction of the car insurance claim, development of several questions was needed and the development of these questions was done based upon the information that was gathered from several sources.

The information gathered had helped us understand what could be the essential factors that can be used in prediction of the car insurance. Several features were also developed during the research for the questions. Initially the research questions only included the features of the car, safety rating, make and model. During the process of research several other factors were also included such as the area of the cluster, population density in that area of the cluster, type of car segments.

All these questions were considered and should be studied, answered, factored in the prediction of whether a policyholder will make a claim in the next six months.

**DATASET**

The dataset used in this project, titled "Car Insurance Claim Prediction" by iftesha najnin, is sourced from kaggle.com and it contains of 44 features and 58,592 instances about policyholders and their cars such as

**policy\_id:**

It is a unique identifier of the policyholder.

**policy\_tenure:**

It describes the time period of the policy.

**age\_of\_car:**

It describes the normalized age of the car in years

**age\_of\_policyholder:**

It describes the normalized age of policyholder in years

**area\_cluster:**

Area cluster of the policyholder

**population density:**

Population density of the policyholder city

**make:**

Encoded Manufacturer/company of the car

**segment:**

Segment of the car (A/ B1/ B2/ C1/ C2)

**model:**

Encoded name of the car

**fuel\_type:**

Type of fuel used by the car

**max\_torque:**

Maximum Torque generated by the car

**max\_power:**

Maximum Power generated by the car

**engine\_type:**

Type of engine used in the car

**airbags:**

Number of airbags installed in the car.

**is\_esc:**

Boolean flag indicating whether Electronic Stability Control (ESC) is present in the car or not.

**is\_adjustable\_steering:**

Boolean flag indicating whether the steering wheel of the car is adjustable or not.

**is\_tpms:**

Boolean flag indicating whether Tire Pressure Monitoring System (TPMS) is present in the car or not.

**is\_parking\_sensors:**

Boolean flag indicating whether parking sensors are present in the car or not.

**is\_parking\_camera:**

Boolean flag indicating whether the parking camera is present in the car or not.

**rear\_brakes\_type:**

Type of brakes used in the rear of the car

**displacement:**

Engine displacement of the car (cc)

**cylinder:**

Number of cylinders present in the engine of the car

**transmission\_type:**

Transmission type of the car.

**gear\_box:**

Number of gears in the car

**steering\_type:**

Type of the power steering present in the car

**turning\_radius:**

The space a vehicle needs to make a certain turn in meters

**length:**

The Length of the car in millimeter.

**width:**

The Width of the car in millimeter

**height:**

The Height of the car in millimeter

**gross\_weight:**

The maximum allowable weight of the fully-loaded car, including passengers, cargo and equipment in kilograms

**is\_front\_fog\_lights:**

It is a Boolean flag indicating whether front fog lights are available in the car or not*.*

**is\_rear\_window\_wiper:**

It is a Boolean flag indicating whether the rear window wiper is available in the car or not*.*

**is\_rear\_window\_washer:**

It is a Boolean flag indicating whether the rear window washer is available in the car or not.

**is\_rear\_window\_defogger:**

It is a Boolean flag indicating whether rear window defogger is available in the car or not.

**is\_brake\_assist:**

It is a Boolean flag indicating whether the brake assistance feature is available in the car or not*.*

**is\_power\_door\_lock:**

It is a Boolean flag indicating whether a power door lock is available in the car or not.

**is\_central\_locking:**

It is a Boolean flag indicating whether the central locking feature is available in the car or not.

**is\_power\_steering:**

It is a Boolean flag indicating whether power steering is available in the car or not.

**is\_driver\_seat\_height\_adjustable:**

It is a Boolean flag indicating whether the height of the driver seat is adjustable or not*.*

**is\_day\_night\_rear\_view\_mirror:**

It is a Boolean flag indicating whether day & night rearview mirror is present in the car or not*.*

**is\_ecw:**

It is a Boolean flag indicating whether Engine Check Warning (ECW) is available in the car or not*.*

**is\_speed\_alert:**

It is a Boolean flag indicating whether the speed alert system is available in the car or not.

**ncap\_rating:**

It is the safety rating given by NCAP (out of 5)

**is\_claim:**

It is the target variable indicating whether the policyholder files a claim in the next six months or not.

**DATA ANALYSIS**

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**from** **tqdm.auto** **import** tqdm

**from** **sklearn** **import** preprocessing

**import** **matplotlib.pyplot** **as** **plt**

In [ ]:

Various libraries are used in this project such as pandas, which is a popular open-source Python library for data science, data analysis, and machine learning. It is constructed on top of the multi-dimensional array-supporting Numpy library. As one of the most well-liked data-wrangling software, Pandas is widely used. Pandas is typically included in every Python installation and integrates nicely with a wide variety of other data science modules within the Python ecosystem.

The second library that is used is Numpy, nearly all branches of research and engineering use the free source Python library known as Numerical Python. It is the dominant approach for manipulating numerical data in Python and forms the basis of both the Py-data and scientific Python ecosystems. Data structures for multi-dimensional arrays and matrices are available in the NumPy library. Arrays can be subjected to a wide range of mathematical operations using NumPy.

The third library that is used is seaborn, The Python data structure pandas is tightly connected with the data visualization package Seaborn, which was constructed on top of matplotlib. The core of Seaborn is a visual representation, which aids in comprehension and exploration of data.

Tqdm auto, sklearn, and matplotlib are the other three libraries that are used. Tqdm is used to produce progress bars and progress meters. At the same time, sklearn is used for machine learning since it contains statistical modeling and machine learning tools such as regression, classification, and clustering. Matplotlib library is used for plotting graphs. It is mostly used for the visualization of data. It is a free alternative to MATLAB.

df = pd.read\_csv(‘train.csv’)

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**from** **tqdm.auto** **import** tqdm

**from** **sklearn** **import** preprocessing

**import** **matplotlib.pyplot** **as** **plt**

In [ ]:

df.head()

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**from** **tqdm.auto** **import** tqdm

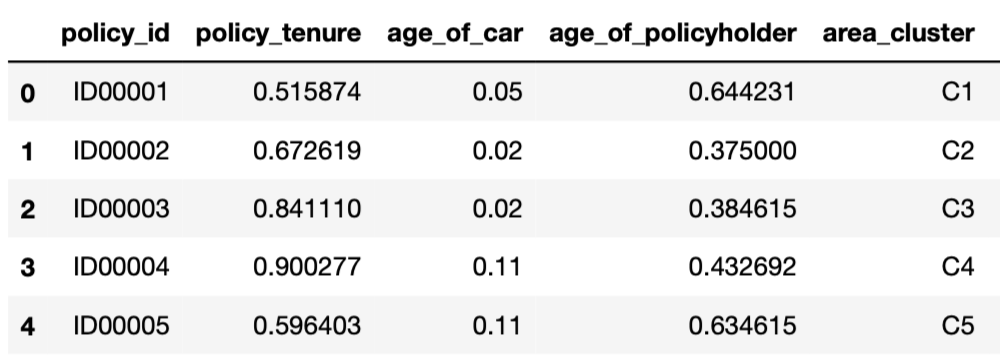
**from** **sklearn** **import** preprocessing

**import** **matplotlib.pyplot** **as** **plt**

In [ ]:

The data set file that contains all the 44 features and 58,592 instances “train.csv” is read and its data frame is created.

Out [ ]:



**Output of the data frame (1.a)**

The above data frame displays 5 instances by default since no value is given in particular. It also shows all the features such as policy\_id, policy\_tenure, age\_of\_car, age\_of\_policyholder, area\_cluster.

numerical= df.select\_dtypes(include =[np.float64,np.int64])

print("Numerical Features in DataSet:",numerical.shape[1])

print(numerical.columns)

In [ ]:

Numerical Features in DataSet: 16

Index(['policy\_tenure', 'age\_of\_car', 'age\_of\_policyholder',

'population\_density', 'make', 'airbags', 'displacement', 'cylinder',

'gear\_box', 'turning\_radius', 'length', 'width', 'height',

'gross\_weight', 'ncap\_rating', 'is\_claim'],

dtype='object')

Using the above code sixteen numerical features were distinguished from the quantitative features. Hence from all the 44 features, there are twenty-eight qualitative features and sixteen quantitative features.

In [ ]:

bar\_claim\_values = df.is\_claim.value\_counts()

print(bar\_claim\_values)

**import** **seaborn** **as** **sns**

**from** **tqdm.auto** **import** tqdm

**from** **sklearn** **import** preprocessing

**import** **matplotlib.pyplot** **as** **plt**

0 54844

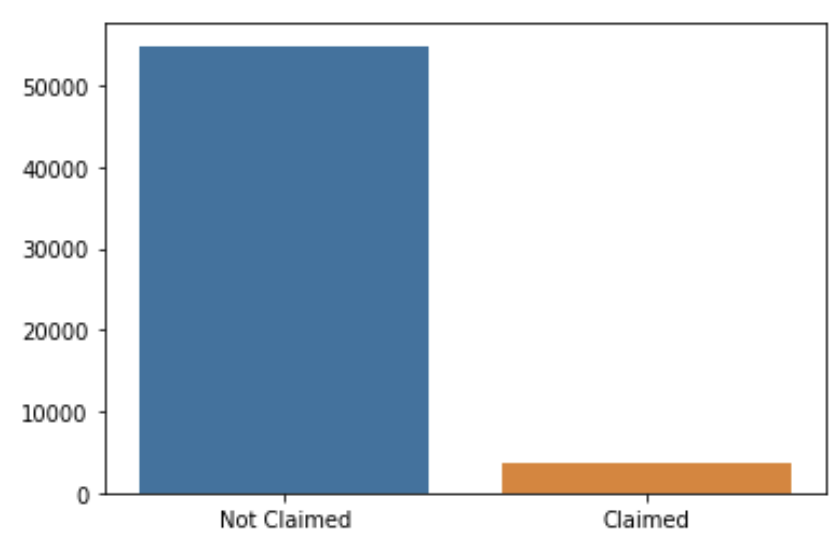
1 3748

Name: is\_claim, dtype: int64

fig = sns.barplot(x = [“Not Claimed”, “Claimed”], y = [bar\_claim\_values[0],bar\_claim\_values[1]])

print(bar\_claim\_values)

In [ ]:



**Bar graph diagram (1.b)**

The above code contains sns.barplot function, which calculates a summary statistic for each category and plots a bar graph. The above bar graph is the visualization of the output of the number of policyholders who made a claim in the first six months and the number of policyholders who did not claim in the first six months. As can be observed from the bar graph, more policyholders made a claim in the first six months than those who didn’t.

**for** columns **in** tqdm(df.colums):

**if** dict(df.dtypes)[columns] == ‘object’:

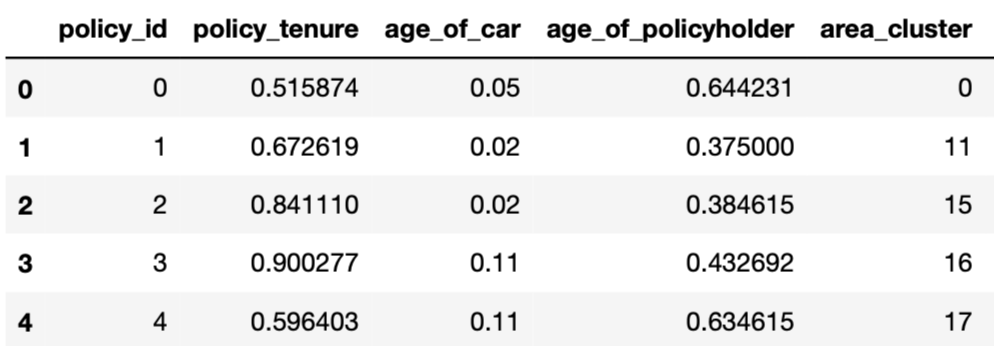
label\_encoder = preprocessing.LabelEncoder()

df[columns]label\_encoder.fit\_transform(df[columns])

In [ ]:

df.head()

In [ ]:

Out [ ]:

**Output of the data frame (2.a)**

By using the label encoding process with the help of the label encoder function from the sklearn library. Which normalizes the data or converts the labels into a numerical form. In our code it converts all the categorical variables into numerical form.

corr\_matrix = df.corr().abs()

print(corr\_matrix)

upper\_tri= corr\_matrix.where(np.triu(np.ones(corr\_matrix.shape),k=1).astype(np.bool))

print(upper\_tri)

In [ ]:

In the above code by using the correlation matrix function which usually calculates Pearson correlation coefficient between input columns of each pair, the correlation between all the forty-four features is calculated.

plt.figure(figsize=(20,15))

heat\_map = sns.heatmap(df.corr(),annot = **True**, fmt = '0.1f')

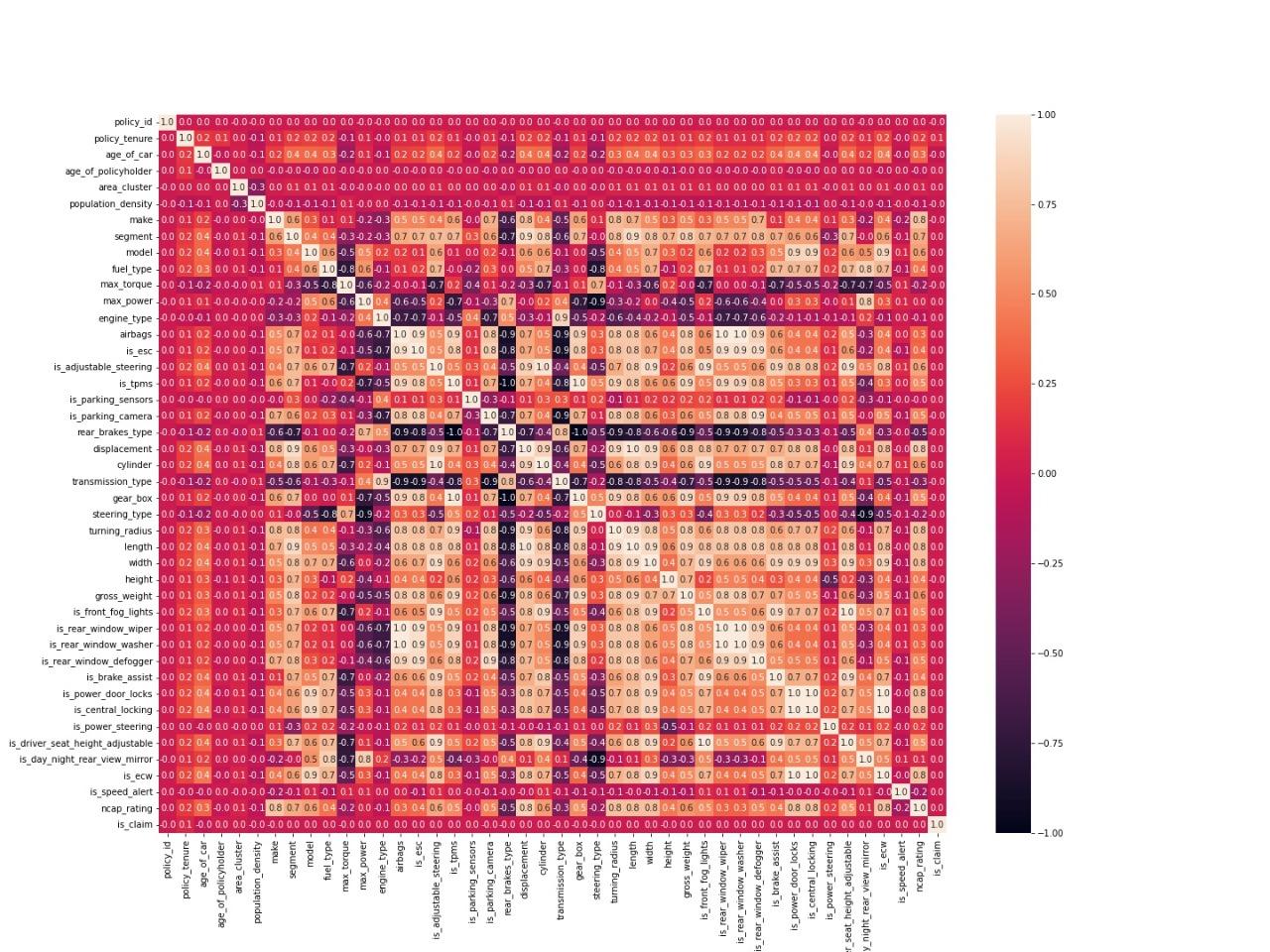
plt.savefig("heat\_map.jpg")

heat\_map

In [ ]:

Out[ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f3158a060a0>

Using the heat map function from the sklearn library which is usually used to visualize the correlation strength among the variables. A heatmap is generated that provides the visualization of correlation between the all the forty-fore variables, which includes sixteen quantitative and twenty-eight qualitative variables or features of the data set.



**Heat map correlation diagram (2.b)**

In [ ]:

to\_drop = [column **for** column in upper\_tri.columns if any(upper\_tri[column] > 0.95)]

print(to\_drop)

[‘rear\_brakes\_type’, ‘cylinder’, ‘gear\_box’, ‘length’, ‘is\_rear\_window\_wiper’, ‘is\_rear\_window\_washer’, ‘is\_central\_locking’, ‘is\_driver\_seat\_height\_adjustable’, ‘is\_ecw’]

The above code is used to pick the columns that contain features with highest correlation, which is usually greater than 95 percent and after those columns with highest correlation are picked to be dropped.

df1 = df.drop(df[to\_drop], axis=1)

print();

print(df1.head())

In [ ]:

Using the to-drop function, all the columns containing the highest correlation features are dropped. After the dropping, the highly correlative features remained at thirty-five (35), which was forty-four initially.

area\_and\_claim = pd.crosstab(df[‘area\_cluster’],df[‘is\_claim’])

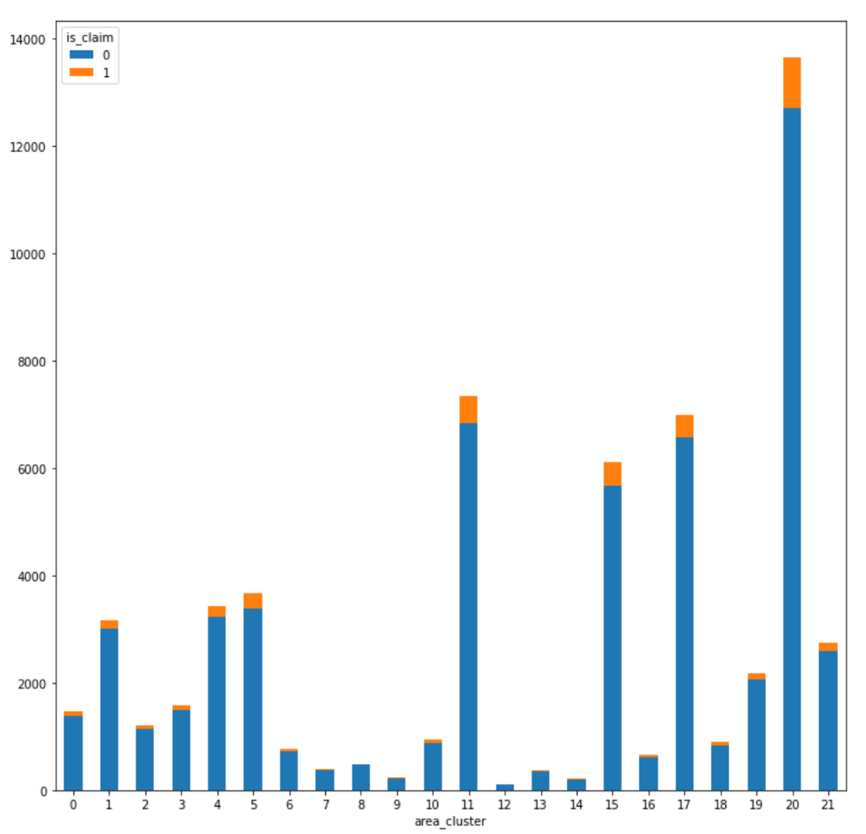
area\_and\_claim\_with\_percent\_claim = area\_and\_claim.apply(**lambda** r:r/r.sum(), axis=1)

In [ ]:

area\_and\_claim.plot(kind=”bar”, stacked=**True**,rot=0, figsize=(12, 1, 2))

In [ ]:

Out [ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fc4a6ffd0>

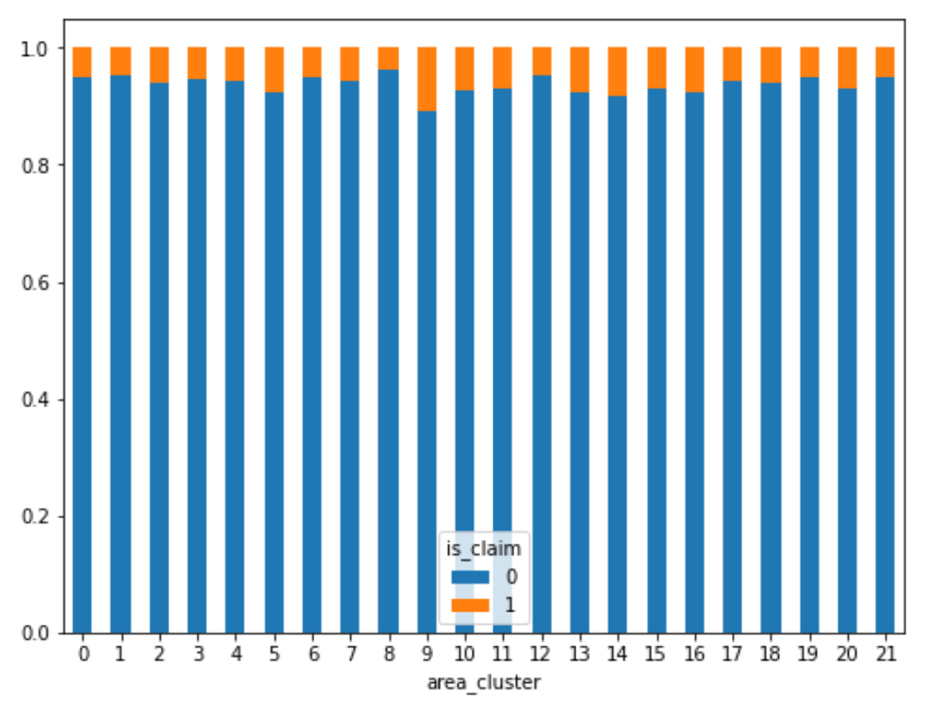


**Bar graph of area\_of\_cluster diagram (3.a)**

area\_and\_claim\_with\_percent\_claim.plot(kind=”bar”, stacked=**True**, ro t=0, figsize=(8, 6))

In [ ]:

Out [ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fc19c5550>



**Bar graph of area\_of\_cluster diagram (3.b)**

From the Bar graph of area of cluster. It can be observed that compared to 21 area of clusters, the area cluster 9 has the highest number of claims within the first six months of the policy tenure.

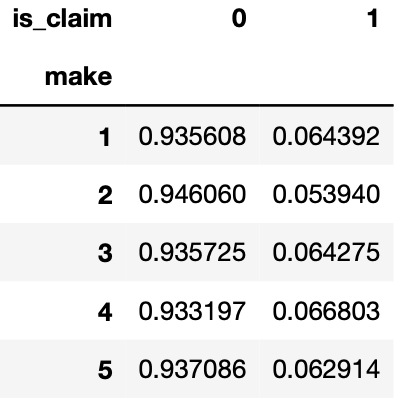
make\_and\_claim = pd.crosstab(df[‘make’],df[‘is\_claim’])

make\_and\_claim\_percent = make\_and\_claim.apply(**lambda** r: r/r.sum(), axis=1)

In [ ]:

make\_and\_claim\_percent

In [ ]:

Out [ ]:

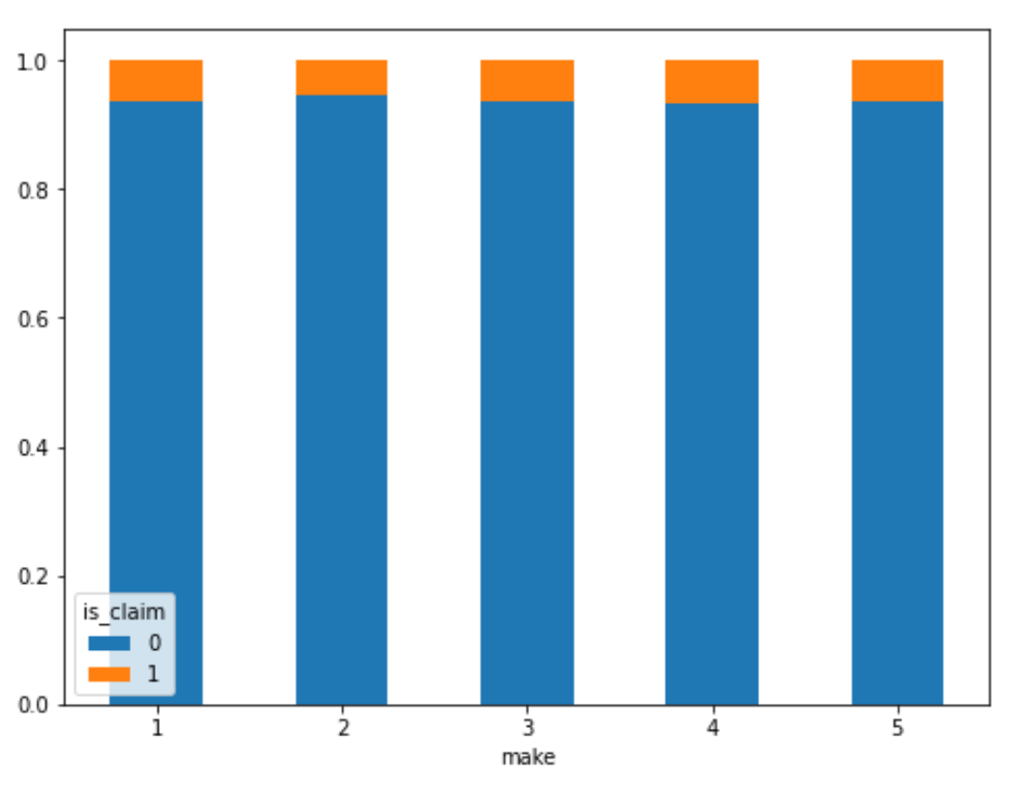
**Output of the make and claim frame (3.c)**

make\_and\_claim\_percent.plot(kind =”bar”, stacked=**True**, rot=0, figsize=(8, 6))

plt.savefig(“make\_and\_claim.jpg”)

In [ ]:

Out [ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fbf8d5f90>



**Bar graph of make and claim (4.a)**

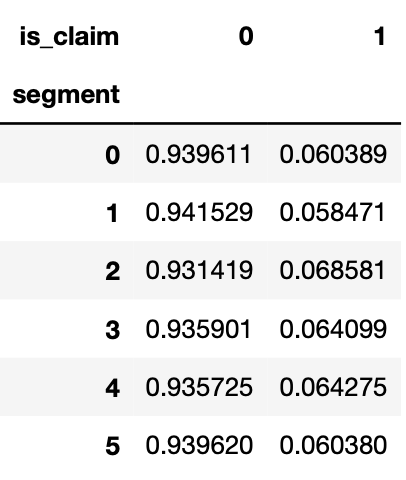
From the above bar graph of make and claim it can be observed that out of the five different makers of cars, the fourth car maker has slightly higher claims than the remaining makers.

seg\_and\_claim = pd.crosstab(df['segment'],df['is\_claim'])

seg\_and\_claim\_percent = seg\_and\_claim.apply(**lambda** r: r/r.sum(), axis=1)

seg\_and\_claim\_percent

In [ ]:

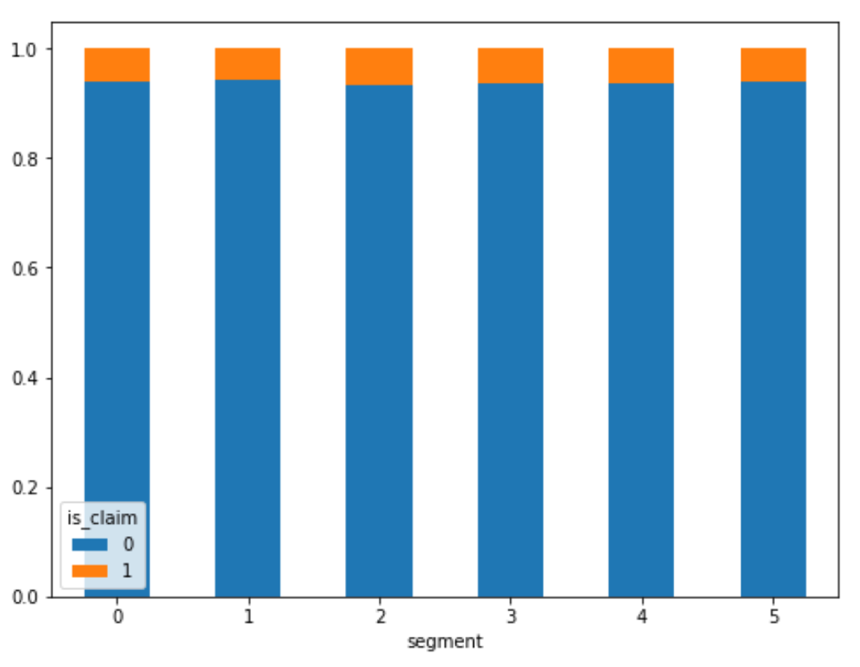
Out [ ]:

**Output of the seg and claim (4.b)**

seg\_and\_claim\_percent.plot(kind="bar", stacked=**True**,rot=0, figsize=(8, 6))

In [ ]:

Out [ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fbf8891d0>



**Bar graph of segment and claim (4.c)**

From the Bar Graph of segment of cars and claim it can be observed that out of the six segment of cars. Segment two cars have slightly higher claims than other four segment of cars.

ncap\_clus\_and\_claim = pd.crosstab([df['ncap\_rating'],df['area\_cluster']],df['is\_claim'])

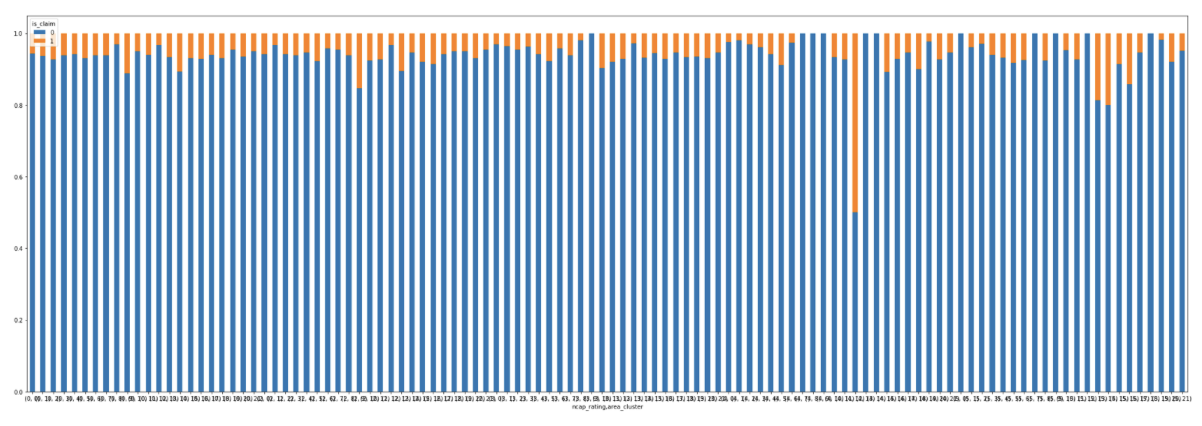
ncap\_clus\_and\_claim\_percent = ncap\_clus\_and\_claim.apply(**lambda** r: r/r.sum(), axis=1)

In [ ]:

ncap\_clus\_and\_claim\_percent.plot(kind="bar", stacked=**True**,rot=0, figsize=(36, 12))

In [ ]:

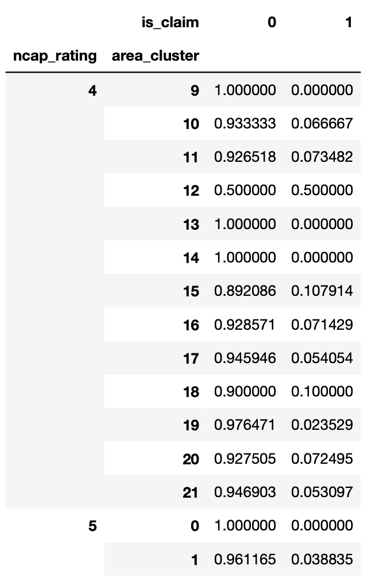
Out [ ]:<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fbfbda8b7d0>



**Bar graph of ncap cluster and claim (5.a)**

ncap\_clus\_and\_claim\_percent[75:90]

In [ ]:



Out [ ]:

**Output of the ncap cluster and claim (5.b)**

From the above graph and Output of the ncap cluster and claim. It can be observed that cars with ncap rating 4 compared to all other rating, in area cluster 12 compared to all other cluster areas have the highest number of claims.

df1.columns

In [ ]:

Out [ ]:

Index(['policy\_id', 'policy\_tenure', 'age\_of\_car', 'age\_of\_policyholder',

'area\_cluster', 'population\_density', 'make', 'segment', 'model',

'fuel\_type', 'max\_torque', 'max\_power', 'engine\_type', 'airbags',

'is\_esc', 'is\_adjustable\_steering', 'is\_tpms', 'is\_parking\_sensors',

'is\_parking\_camera', 'displacement', 'transmission\_type',

'steering\_type', 'turning\_radius', 'width', 'height', 'gross\_weight',

'is\_front\_fog\_lights', 'is\_rear\_window\_defogger', 'is\_brake\_assist',

'is\_power\_door\_locks', 'is\_power\_steering',

'is\_day\_night\_rear\_view\_mirror', 'is\_speed\_alert', 'ncap\_rating',

'is\_claim'],

dtype='object')

age\_and\_claim = pd.crosstab(df1['age\_of\_policyholder'],df1['is\_claim'],rownames=['age\_of\_policyholder'])

In [ ]:

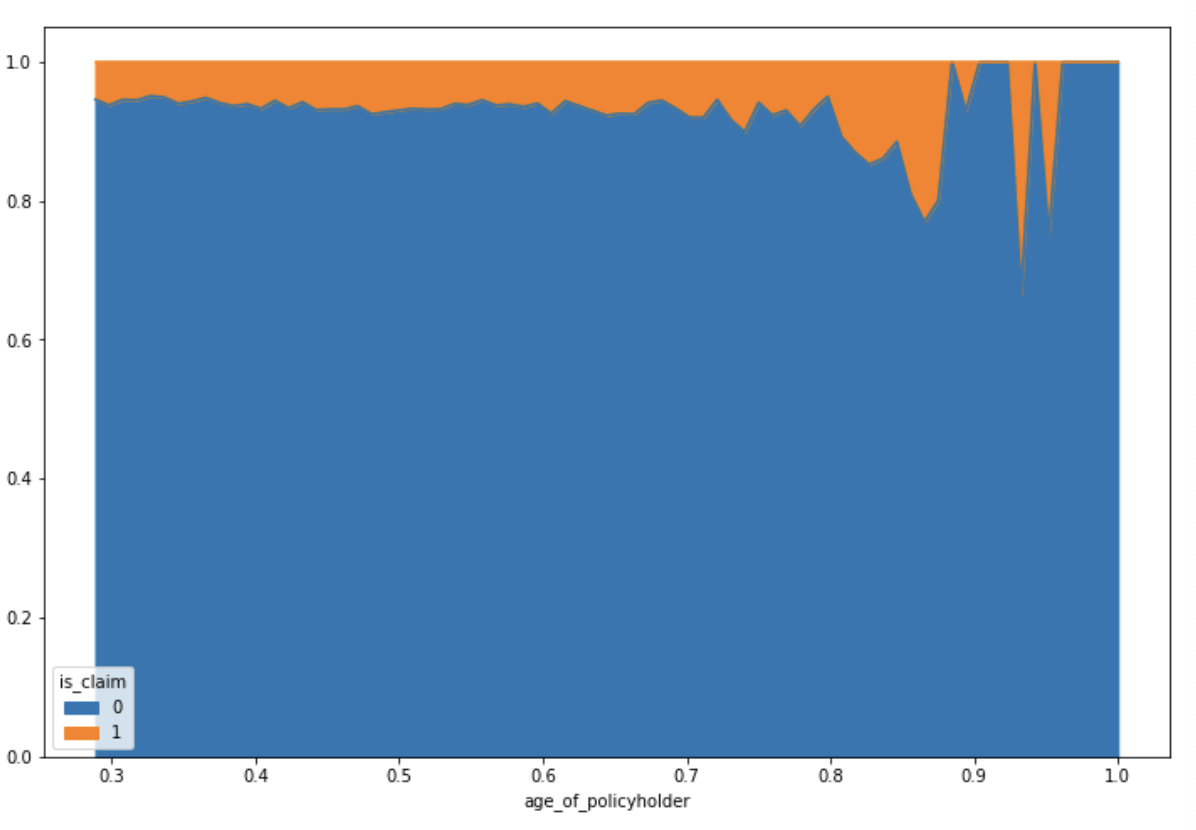
age\_and\_claim\_percent = age\_and\_claim.apply(**lambda** r: r/r.sum(), axis=1)

age\_and\_claim\_percent\_fig = age\_and\_claim\_percent.plot.area(figsize = (12,8))

age\_and\_claim\_percent\_fig

In [ ]:

Out [ ]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fbdla5c50>

****

**Graph plot of age of policyholder and claim (5.c)**

From the above Graph plot of age of policyholder and claim it can be observed that people of old age or older people tend to make more claims than people of young age or young ones.

sorted(df['age\_of\_policyholder'].unique())

In [ ]:

age\_car\_and\_claim = pd.crosstab(df1['age\_of\_car'],df['is\_claim'])

age\_car\_and\_claim\_percent = age\_car\_and\_claim.apply(**lambda** r: r/r.sum(), axis=1)

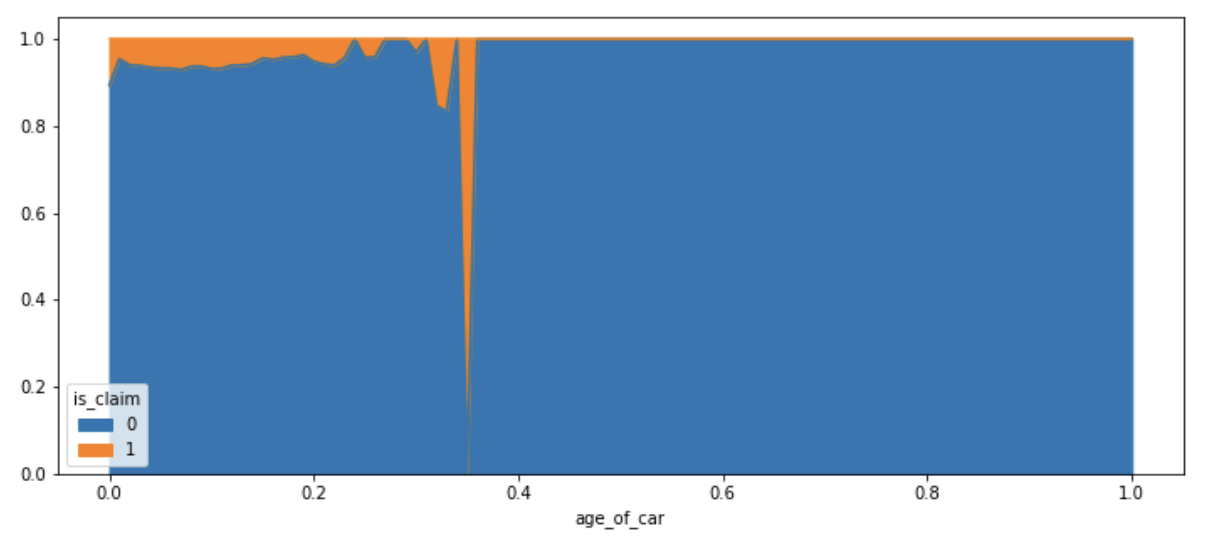
age\_car\_and\_claim\_percent\_fig = age\_car\_and\_claim\_percent.plot.area(figsize = (12,5))

age\_car\_and\_claim\_percent\_fig

In [ ]:

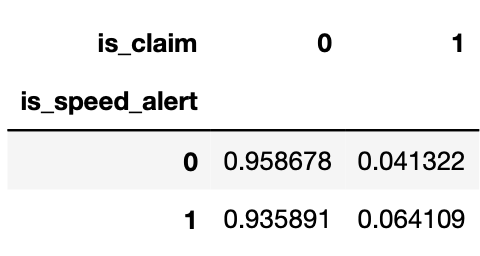
Out [ ]:<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4fbd03dd10>

**Graph plot of age of car and claim (6.a)**

****By observing the above graph plot of age of car and claim. Compared to policyholders of the older cars, the policy holders of the newer cars tend to make more claims.

pd.crosstab(df1['is\_speed\_alert'],df1['is\_claim']).apply(**lambda** r: r/r.sum(), axis=1)

In [ ]:

****

Out [ ]:

**Output of the speed alert and claim (6.b)**

ffg = df.groupby("is\_claim")["segment"].value\_counts().rename("Frequency").to\_frame().reset\_index()

sns.barplot(x="is\_claim",

y="Frequency",

hue="segment",

data=ffg)

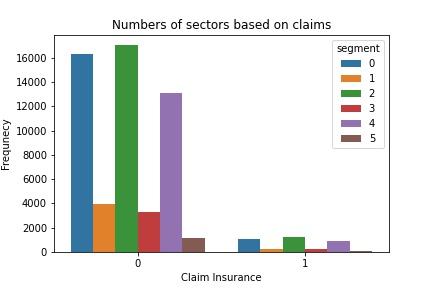
plt.xlabel("Claim Insurance")

plt.ylabel("Frequnecy")

plt.title("Numbers of sectors based on claims")

plt.savefig("claim\_and\_segment.jpg")

In [ ]:



**Bar graph of cars segment and type of claims (6.c)**

Using the sns barplot which is usually used to calculate a summary statistic mean for each category, plots a bar graph and dataframe group by function which is used to split the data into groups based on certain conditons. Comparision between cars segment and type of claims can be observed.

In [ ]:

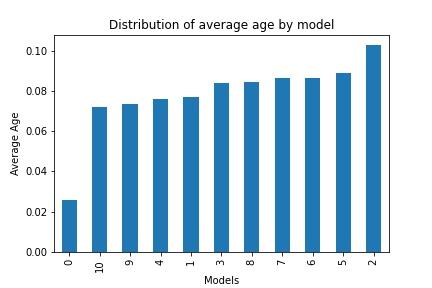
df.groupby("model")["age\_of\_car"].mean().sort\_values().plot(kind="bar")

plt.xlabel("Models")

plt.ylabel("Average Age")

plt.title("Distribution of average age by model");

plt.savefig("model\_and\_ageofcar.jpg")



**Bar graph of Models and Average age (7.a)**

Using the dataframe groupby function which is used to split the data into groups based on specific attributes. Distribution of the average age segment wise can be observed.

**RESULT**

During the analysis several observations were made and these observations answered the research questions. The first research question was “which area cluster has more insurance claims?” It was observed during the analysis that out of twenty-one area clusters, the area cluster 9 has the highest percent of claims within the first six months of the policy tenure.

Following that the second research question was “Which people of age driving the cars are claiming the insurance?” It was noticed during the analysis that people of older age or older people tend to make more claims than the people of young age or younger ones. The third research question was “What model and make of cars used by the people in the cluster claim insurance?” It was also observed during the analysis that the fourth car maker has the slightly higher claims than the remaining makers.

The fourth research question was “Owners of which car segment are claiming the insurance the most?” It was perceived in the analysis that cars of segment two have slightly higher claims than others. The fifth research question was “Cars with what safety rating in an area of the cluster has a high number of claims?” It was also observed during the analysis that cars with the ncap safety rating four in the area of cluster twelve have the highest number of claims. The last two research questions were “Owners of cars with what type of features are claiming insurance? And How many non-corelative features of the dataset factor in for the outcome of the prediction?” There were initially about forty-four features in the data set which also include few features related to the car. In these forty-four features, there are sixteen quantitative features and twenty-eight qualitative features, nine of the sixteen numerical features are related to cars. After correlating, heat-mapping and dropping the highly correlative features there were thirty-five (35) non-correlative features that were factored in during the analysis.

**CONCLUSION**

As expected we understood the factors that are related to the analysis of whether a policyholder will file a claim in the next six months and with the help of the observations during the analysis, all the research questions were answered. The strengths of this project were the necessity of the high number of instances during the process of analysis and the limitations of this project were few non-correlative features. The future of this project can extend from analysing the factors of predicting the claim of insurance in the next six months to analysing the factors of predicting the claim of insurance in the next month

**REFERENCES**

<https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification?select=sample_submission.csv>

<https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>

<https://numpy.org/doc/stable/user/absolute_beginners.html>

<https://towardsdatascience.com/seaborn-python-8563c3d0ad41>

<https://snyk.io/advisor/python/sklearn/functions/sklearn.preprocessing.LabelEncoder>