In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wr
```

In [7]:
```
df= pd.read_csv("car_insurance_claim[1].csv")
```

In [8]:
```
print(df.head())
```

```
          ID  KIDSDRIV    BIRTH   AGE  HOMEKIDS   YOJ   INCOME PARENT1  \
0   63581743         0  16MAR39  60.0         0  11.0  $67,349      No
1  132761049         0  21JAN56  43.0         0  11.0  $91,449      No
2  921317019         0  18NOV51  48.0         0  11.0  $52,881      No
3  727598473         0  05MAR64  35.0         1  10.0  $16,039      No
4  450221861         0  05JUN48  51.0         0  14.0      NaN      No

   HOME_VAL MSTATUS  ... CAR_TYPE RED_CAR OLDCLAIM  CLM_FREQ REVOKED MVR_PTS  \
0        $0    z_No  ...  Minivan     yes   $4,461         2      No       3
1  $257,252    z_No  ...  Minivan     yes       $0         0      No       0
2        $0    z_No  ...      Van     yes       $0         0      No       2
3  $124,191     Yes  ...    z_SUV      no  $38,690         2      No       3
4  $306,251     Yes  ...  Minivan     yes       $0         0      No       0

   CLM_AMT CAR_AGE CLAIM_FLAG           URBANICITY
0       $0    18.0          0  Highly Urban/ Urban
1       $0     1.0          0  Highly Urban/ Urban
2       $0    10.0          0  Highly Urban/ Urban
3       $0    10.0          0  Highly Urban/ Urban
4       $0     6.0          0  Highly Urban/ Urban

[5 rows x 27 columns]
```

In [9]:
```
df.head()
```

Out[9]:

	ID	KIDSDRIV	BIRTH	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HOME_VAL	MSTATU
0	63581743	0	16MAR39	60.0	0	11.0	$67,349	No	$0	z_N
1	132761049	0	21JAN56	43.0	0	11.0	$91,449	No	$257,252	z_N
2	921317019	0	18NOV51	48.0	0	11.0	$52,881	No	$0	z_N
3	727598473	0	05MAR64	35.0	1	10.0	$16,039	No	$124,191	Ye
4	450221861	0	05JUN48	51.0	0	14.0	NaN	No	$306,251	Ye

5 rows × 27 columns

In [10]:

```
df.shape
```

Out[10]:

```
(10302, 27)
```

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10302 entries, 0 to 10301
Data columns (total 27 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   ID          10302 non-null  int64  
 1   KIDSDRIV    10302 non-null  int64  
 2   BIRTH       10302 non-null  object 
 3   AGE         10295 non-null  float64
 4   HOMEKIDS    10302 non-null  int64  
 5   YOJ         9754 non-null   float64
 6   INCOME      9732 non-null   object 
 7   PARENT1     10302 non-null  object 
 8   HOME_VAL    9727 non-null   object 
 9   MSTATUS     10302 non-null  object 
 10  GENDER      10302 non-null  object 
 11  EDUCATION   10302 non-null  object 
 12  OCCUPATION  9637 non-null   object 
 13  TRAVTIME    10302 non-null  int64  
 14  CAR_USE     10302 non-null  object 
 15  BLUEBOOK    10302 non-null  object 
 16  TIF         10302 non-null  int64  
 17  CAR_TYPE    10302 non-null  object 
 18  RED_CAR     10302 non-null  object 
 19  OLDCLAIM    10302 non-null  object 
 20  CLM_FREQ    10302 non-null  int64  
 21  REVOKED     10302 non-null  object 
 22  MVR_PTS     10302 non-null  int64  
 23  CLM_AMT     10302 non-null  object 
 24  CAR_AGE     9663 non-null   float64
 25  CLAIM_FLAG  10302 non-null  int64  
 26  URBANICITY  10302 non-null  object 
dtypes: float64(3), int64(8), object(16)
memory usage: 2.1+ MB
```

In [12]:
```
df.describe()
```

Out[12]:

	ID	KIDSDRIV	AGE	HOMEKIDS	YOJ	TRAVTIME	
count	1.030200e+04	10302.000000	10295.000000	10302.000000	9754.000000	10302.000000	10302.0000
mean	4.956631e+08	0.169288	44.837397	0.720443	10.474062	33.416424	5.3291
std	2.864675e+08	0.506512	8.606445	1.116323	4.108943	15.869687	4.1107
min	6.317500e+04	0.000000	16.000000	0.000000	0.000000	5.000000	1.0000
25%	2.442869e+08	0.000000	39.000000	0.000000	9.000000	22.000000	1.0000
50%	4.970043e+08	0.000000	45.000000	0.000000	11.000000	33.000000	4.0000
75%	7.394551e+08	0.000000	51.000000	1.000000	13.000000	44.000000	7.0000
max	9.999264e+08	4.000000	81.000000	5.000000	23.000000	142.000000	25.0000

In [13]:
```
df.isnull().sum()
```

Out[13]:

```
ID              0
KIDSDRIV        0
BIRTH           0
AGE             7
HOMEKIDS        0
YOJ           548
INCOME        570
PARENT1         0
HOME_VAL      575
MSTATUS         0
GENDER          0
EDUCATION       0
OCCUPATION    665
TRAVTIME        0
CAR_USE         0
BLUEBOOK        0
TIF             0
CAR_TYPE        0
RED_CAR         0
OLDCLAIM        0
CLM_FREQ        0
REVOKED         0
MVR_PTS         0
CLM_AMT         0
CAR_AGE       639
CLAIM_FLAG      0
URBANICITY      0
dtype: int64
```

In [ ]:

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
ID              0
KIDSDRIV        0
BIRTH           0
AGE             7
HOMEKIDS        0
YOJ           548
INCOME        570
PARENT1         0
HOME_VAL      575
MSTATUS         0
GENDER          0
EDUCATION       0
OCCUPATION    665
TRAVTIME        0
CAR_USE         0
BLUEBOOK        0
TIF             0
CAR_TYPE        0
RED_CAR         0
OLDCLAIM        0
CLM_FREQ        0
REVOKED         0
MVR_PTS         0
CLM_AMT         0
CAR_AGE       639
CLAIM_FLAG      0
URBANICITY      0
dtype: int64
```

In [15]:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
encoded_data = label_encoder.fit_transform(df['CLM_AMT'])
print(encoded_data)
```

```
[0 0 0 ... 0 0 0]
```

In [ ]:

In [16]:

```
df.columns
```

Out[16]:

```
Index(['ID', 'KIDSDRIV', 'BIRTH', 'AGE', 'HOMEKIDS', 'YOJ', 'INCOME',
       'PARENT1', 'HOME_VAL', 'MSTATUS', 'GENDER', 'EDUCATION', 'OCCUPATION',
       'TRAVTIME', 'CAR_USE', 'BLUEBOOK', 'TIF', 'CAR_TYPE', 'RED_CAR',
       'OLDCLAIM', 'CLM_FREQ', 'REVOKED', 'MVR_PTS', 'CLM_AMT', 'CAR_AGE',
       'CLAIM_FLAG', 'URBANICITY'],
      dtype='object')
```

In [17]:

```
df['CLM_AMT']=df['CLM_AMT'].value_counts()
```

In [18]:

```
df.describe()
```

Out[18]:

	ID	KIDSDRIV	AGE	HOMEKIDS	YOJ	TRAVTIME	·
count	1.030200e+04	10302.000000	10295.000000	10302.000000	9754.000000	10302.000000	10302.0000
mean	4.956631e+08	0.169288	44.837397	0.720443	10.474062	33.416424	5.3291
std	2.864675e+08	0.506512	8.606445	1.116323	4.108943	15.869687	4.1107
min	6.317500e+04	0.000000	16.000000	0.000000	0.000000	5.000000	1.0000
25%	2.442869e+08	0.000000	39.000000	0.000000	9.000000	22.000000	1.0000
50%	4.970043e+08	0.000000	45.000000	0.000000	11.000000	33.000000	4.0000
75%	7.394551e+08	0.000000	51.000000	1.000000	13.000000	44.000000	7.0000
max	9.999264e+08	4.000000	81.000000	5.000000	23.000000	142.000000	25.0000

In [19]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.metrics import classification_report, mean_squared_error


import numpy as np
import re
import pandas as pd
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [20]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\dell\.conda\acc\lib\site-packages
(2.0.3)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: scipy in c:\users\dell\.conda\acc\lib\site-packages (f
rom xgboost) (1.10.0)
Requirement already satisfied: numpy in c:\users\dell\.conda\acc\lib\site-packages (f
rom xgboost) (1.23.5)
```

In [21]:

```
import xgboost as xgb
```

In [22]:

```
data = pd.read_csv('car_insurance_claim[1].csv')
```

In [23]:

```
data = pd.get_dummies(data)
```

In [24]:
```
ps = PorterStemmer()
def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
    stemmed_content = [ps.stem(word) for word in stemmed_content if not word in stopwo
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

In [25]:
```
X = df.drop('CAR_AGE',axis=1)
y = df['CAR_AGE']
```

In [26]:
```
print(X)
```

```
              ID  KIDSDRIV    BIRTH   AGE  HOMEKIDS   YOJ    INCOME PARENT1  \
0       63581743         0  16MAR39  60.0         0  11.0   $67,349      No   
1      132761049         0  21JAN56  43.0         0  11.0   $91,449      No   
2      921317019         0  18NOV51  48.0         0  11.0   $52,881      No   
3      727598473         0  05MAR64  35.0         1  10.0   $16,039      No   
4      450221861         0  05JUN48  51.0         0  14.0       NaN      No   
...          ...       ...      ...   ...       ...   ...       ...     ...   
10297   67790126         1  13AUG54  45.0         2   9.0  $164,669      No   
10298   61970712         0  17JUN53  46.0         0   9.0  $107,204      No   
10299  849208064         0  18JUN51  48.0         0  15.0   $39,837      No   
10300  627828331         0  12DEC48  50.0         0   7.0   $43,445      No   
10301  680381960         0  27FEB47  52.0         0  11.0   $53,235      No   

        HOME_VAL MSTATUS  ... TIF     CAR_TYPE RED_CAR  OLDCLAIM CLM_FREQ  \
0             $0    z_No  ...  11      Minivan     yes    $4,461        2   
1       $257,252    z_No  ...   1      Minivan     yes        $0        0   
2             $0    z_No  ...   1          Van     yes        $0        0   
3       $124,191     Yes  ...   4        z_SUV      no   $38,690        2   
4       $306,251     Yes  ...   7      Minivan     yes        $0        0   
...          ...     ...  ...  ..          ...     ...       ...      ...   
10297   $386,273     Yes  ...  15      Minivan      no        $0        0   
10298   $332,591     Yes  ...   6  Panel Truck      no        $0        0   
10299   $170,611     Yes  ...   7        z_SUV      no        $0        0   
10300   $149,248     Yes  ...   6      Minivan      no        $0        0   
10301   $197,017     Yes  ...   6      Minivan      no        $0        0   

      REVOKED  MVR_PTS CLM_AMT CLAIM_FLAG            URBANICITY  
0          No        3     NaN          0   Highly Urban/ Urban  
1          No        0     NaN          0   Highly Urban/ Urban  
2          No        2     NaN          0   Highly Urban/ Urban  
3          No        3     NaN          0   Highly Urban/ Urban  
4          No        0     NaN          0   Highly Urban/ Urban  
...       ...      ...     ...        ...                   ...  
10297      No        2     NaN          0   Highly Urban/ Urban  
10298      No        0     NaN          0   Highly Urban/ Urban  
10299      No        0     NaN          0   Highly Urban/ Urban  
10300      No        0     NaN          0   Highly Urban/ Urban  
10301      No        0     NaN          0 z_Highly Rural/ Rural  

[10302 rows x 26 columns]
```

In [27]:
```
df['CLM_AMT'] = df['CLM_AMT'].value_counts().apply(stemming)
```

In [28]:
```
df['AGE']
```

Out[28]:

```
0        60.0
1        43.0
2        48.0
3        35.0
4        51.0
         ...
10297    45.0
10298    46.0
10299    48.0
10300    50.0
10301    52.0
Name: AGE, Length: 10302, dtype: float64
```

In [29]:

```
X_prob = df.drop(['CAR_AGE'],axis=1)
y_prob = df['CAR_AGE']
```

In [30]:

```
X_amt = df.drop(['CLM_AMT'], axis=1)
y_amt = df['CLM_AMT']
```

In [31]:

```
X_prob_train, X_prob_test, y_prob_train, y_prob_test = train_test_split(X_prob, y_prob
```

In [32]:

```
X_amt_train, X_amt_test, y_amt_train, y_amt_test = train_test_split(X_amt, y_amt, test
```

In [46]:

```
X_amt_train.shape
```

Out[46]:

```
(8241, 26)
```

In [ ]:

```
def bar_chart(feature):
    CAR_AGE = df[df['CAR_AGE']==1][feature].value_counts()
    df.plot(kind='bar',stacked=True, figsize=(15,7))

bar_chart('CAR_AGE')
```

In [ ]:

```
X = df['CAR_AGE'].values
y = df['CLM_FREQ'].values
```

In [ ]:

```
print(X,y)
```

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.2, stratify=y,
```

In [ ]:

```
X_train.shape
```

In [ ]:

```
bar_chart('CLM_FREQ')
```

In [ ]:

```
bar_chart('CLM_AMT')
```

In [ ]:

```
bar_chart(df['CAR_AGE'])
```

In [ ]:

```

```