tamrin travel insurance

import libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

import dataset

In [2]:
```
data=pd.read_csv('tamrin decision tree.csv')
data
```

Out[2]:

	Agency	Agency Type	Distribution Channel	Product Name	Claim	Duration	Destination	Net Sales	Commis (in val
0	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	9
1	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	9
2	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	65	AUSTRALIA	-49.5	29
3	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	60	AUSTRALIA	-39.6	23
4	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	79	ITALY	-19.8	11
...	...	...	...	...	...	...	...	...	
63321	JZI	Airlines	Online	Basic Plan	No	111	JAPAN	35.0	12
63322	JZI	Airlines	Online	Basic Plan	No	58	CHINA	40.0	14
63323	JZI	Airlines	Online	Basic Plan	No	2	MALAYSIA	18.0	6
63324	JZI	Airlines	Online	Basic Plan	No	3	VIET NAM	18.0	6
63325	JZI	Airlines	Online	Basic Plan	No	22	HONG KONG	26.0	9

63326 rows × 11 columns

In [3]:
```
data.columns
```

Out[3]:
```
Index(['Agency', 'Agency Type', 'Distribution Channel', 'Product Name',
       'Claim', 'Duration', 'Destination', 'Net Sales', 'Commision (in value)',
       'Gender', 'Age'],
      dtype='object')
```

In [4]:
```
data.iloc[:,1].value_counts()
```

Out[4]:
```
Travel Agency    45869
Airlines         17457
Name: Agency Type, dtype: int64
```

In [5]:
```
data.Agency.value_counts()
```

Out[5]:
```
EPX    35119
CWT     8580
C2B     8267
JZI     6329
SSI     1056
JWT      749
RAB      725
LWC      689
TST      528
KML      392
ART      331
CCR      194
CBH      101
TTW       98
CSR       86
ADM       82
Name: Agency, dtype: int64
```

In [6]:
```
data.iloc[:,2].value_counts()
```

Out[6]:
```
Online     62219
Offline     1107
Name: Distribution Channel, dtype: int64
```

In [7]:
```
data.iloc[:,3].value_counts()
```

Out[7]:
```
Cancellation Plan                      18630
2 way Comprehensive Plan               13158
Rental Vehicle Excess Insurance         8580
Basic Plan                              5469
Bronze Plan                             4049
1 way Comprehensive Plan                3331
Value Plan                              2715
Silver Plan                             2249
Annual Silver Plan                      1423
Ticket Protector                        1056
Travel Cruise Protect                    527
Comprehensive Plan                       364
Gold Plan                                352
24 Protect                               247
Single Trip Travel Protect Gold          204
Premier Plan                             194
Annual Gold Plan                         194
Single Trip Travel Protect Silver        173
Annual Travel Protect Gold               100
Annual Travel Protect Silver              86
Individual Comprehensive Plan             74
Single Trip Travel Protect Platinum       73
Annual Travel Protect Platinum            53
Spouse or Parents Comprehensive Plan      15
Child Comprehensive Plan                   9
Travel Cruise Protect Family               1
Name: Product Name, dtype: int64
```

In [8]:
```
data1=data[(data['Product Name']=='Cancellation Plan') |
           (data['Product Name']=='2 way Comprehensive Plan') |
           (data['Product Name']=='Rental Vehicle Excess Insurance') |
           (data['Product Name']=='Basic Plan') |
           (data['Product Name']=='Bronze Plan') |
           (data['Product Name']=='1 way Comprehensive Plan') |
           (data['Product Name']=='Value Plan') |
           (data['Product Name']=='Silver Plan') |
           (data['Product Name']=='Annual Silver Plan') |
           (data['Product Name']=='Ticket Protector')]
data1.reset_index().drop('index',axis=1)
```

Out[8]:

	Agency	Agency Type	Distribution Channel	Product Name	Claim	Duration	Destination	Net Sales	Commision (in value)
0	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	65	AUSTRALIA	-49.5	29.70
1	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	60	AUSTRALIA	-39.6	23.76
2	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	79	ITALY	-19.8	11.88
3	JZI	Airlines	Online	Value Plan	No	66	UNITED STATES	-121.0	42.35
4	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	47	UNITED STATES	-39.6	23.76
...	...	...	...	...	...	...	...	...	...
60655	JZI	Airlines	Online	Basic Plan	No	111	JAPAN	35.0	12.25
60656	JZI	Airlines	Online	Basic Plan	No	58	CHINA	40.0	14.00
60657	JZI	Airlines	Online	Basic Plan	No	2	MALAYSIA	18.0	6.30
60658	JZI	Airlines	Online	Basic Plan	No	3	VIET NAM	18.0	6.30
60659	JZI	Airlines	Online	Basic Plan	No	22	HONG KONG	26.0	9.10

60660 rows × 11 columns

In [9]:
```
data1.Claim.value_counts()
```

Out[9]:
```
No     59822
Yes      838
Name: Claim, dtype: int64
```

In [10]:
```
data1.Destination.value_counts()
```

Out[10]:
```
SINGAPORE                                   12612
MALAYSIA                                     5423
THAILAND                                     5337
CHINA                                        4708
AUSTRALIA                                    3649
                                            ...
BOSNIA AND HERZEGOVINA                          1
ARMENIA                                         1
GUATEMALA                                       1
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF          1
URUGUAY                                         1
Name: Destination, Length: 146, dtype: int64
```

In [11]:
```
data1.Gender.value_counts()
```

Out[11]:
```
M    8122
F    7463
Name: Gender, dtype: int64
```

pre-processing

handle categorical data

In [12]:
```
data1.dtypes
```

Out[12]:
```
Agency                  object
Agency Type             object
Distribution Channel    object
Product Name            object
Claim                   object
Duration                 int64
Destination             object
Net Sales              float64
Commision (in value)   float64
Gender                  object
Age                      int64
dtype: object
```

In [13]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60660 entries, 2 to 63325
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Agency                60660 non-null  object 
 1   Agency Type           60660 non-null  object 
 2   Distribution Channel  60660 non-null  object 
 3   Product Name          60660 non-null  object 
 4   Claim                 60660 non-null  object 
 5   Duration              60660 non-null  int64  
 6   Destination           60660 non-null  object 
 7   Net Sales             60660 non-null  float64
 8   Commision (in value)  60660 non-null  float64
 9   Gender                15585 non-null  object 
 10  Age                   60660 non-null  int64  
dtypes: float64(2), int64(2), object(7)
memory usage: 5.6+ MB
```

In [14]:
```
data2=data1[data1['Age']>18].reset_index().drop('index',axis=1)
data2
```

Out[14]:

	Agency	Agency Type	Distribution Channel	Product Name	Claim	Duration	Destination	Net Sales	Commision (in value)
0	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	65	AUSTRALIA	-49.5	29.70
1	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	60	AUSTRALIA	-39.6	23.76
2	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	79	ITALY	-19.8	11.88
3	JZI	Airlines	Online	Value Plan	No	66	UNITED STATES	-121.0	42.35
4	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	47	UNITED STATES	-39.6	23.76
...	...	...	...	...	...	...	...	...	...
60543	JZI	Airlines	Online	Basic Plan	No	111	JAPAN	35.0	12.25
60544	JZI	Airlines	Online	Basic Plan	No	58	CHINA	40.0	14.00
60545	JZI	Airlines	Online	Basic Plan	No	2	MALAYSIA	18.0	6.30
60546	JZI	Airlines	Online	Basic Plan	No	3	VIET NAM	18.0	6.30
60547	JZI	Airlines	Online	Basic Plan	No	22	HONG KONG	26.0	9.10

60548 rows × 11 columns

In [15]:
```
from sklearn import preprocessing
data3=data2.drop(['Gender','Agency','Destination'],axis=1)

label_encoder1 = preprocessing.LabelEncoder()
data3['Distribution Channel']= label_encoder1.fit_transform(data3['Distribution C

label_encoder2 = preprocessing.LabelEncoder()
data3['Claim']= label_encoder2.fit_transform(data3['Claim'])

data4=pd.get_dummies(data3)
data4
```

Out[15]:

	Distribution Channel	Claim	Duration	Net Sales	Commision (in value)	Age	Agency Type_Airlines	Agency Type_Travel Agency	P Name Compreh
0	1	0	65	-49.5	29.70	32	0	1	
1	1	0	60	-39.6	23.76	32	0	1	
2	1	0	79	-19.8	11.88	41	0	1	
3	1	0	66	-121.0	42.35	44	1	0	
4	1	0	47	-39.6	23.76	32	0	1	
...	...	...	...	...	...	...	...	...	
60543	1	0	111	35.0	12.25	31	1	0	
60544	1	0	58	40.0	14.00	40	1	0	
60545	1	0	2	18.0	6.30	57	1	0	
60546	1	0	3	18.0	6.30	63	1	0	
60547	1	0	22	26.0	9.10	35	1	0	

60548 rows × 18 columns

handle missing value

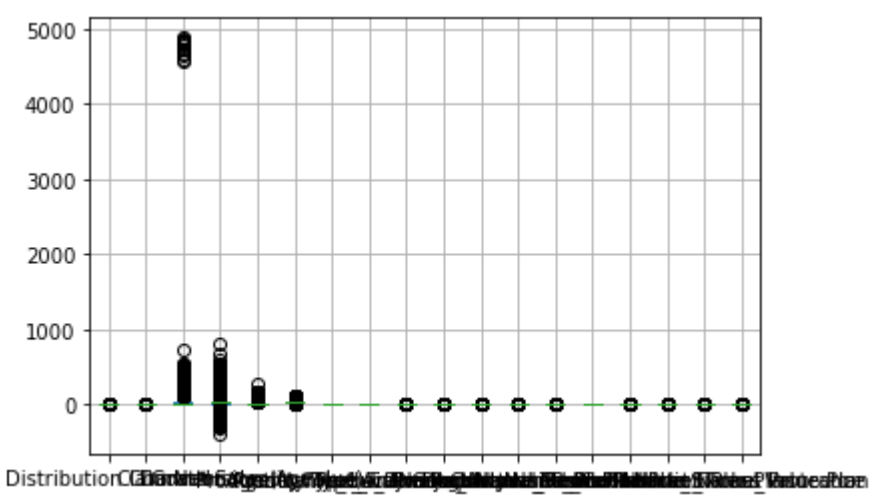
In [16]:
```
data4.isna().sum()
```

Out[16]:
```
Distribution Channel                            0
Claim                                           0
Duration                                        0
Net Sales                                       0
Commision (in value)                            0
Age                                             0
Agency Type_Airlines                            0
Agency Type_Travel Agency                       0
Product Name_1 way Comprehensive Plan           0
Product Name_2 way Comprehensive Plan           0
Product Name_Annual Silver Plan                 0
Product Name_Basic Plan                         0
Product Name_Bronze Plan                        0
Product Name_Cancellation Plan                  0
Product Name_Rental Vehicle Excess Insurance    0
Product Name_Silver Plan                        0
Product Name_Ticket Protector                   0
Product Name_Value Plan                         0
dtype: int64
```

handle outlier data

In [17]:
```
data4.boxplot()
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17e02db8cd0>
```



In [18]:
```
#Duration
Q1=data4.iloc[:,2].quantile(0.25)
Q3=data4.iloc[:,2].quantile(0.75)
LB=Q1-1.5*(Q3-Q1)
UB=Q3+1.5*(Q3-Q1)
print(LB,UB)
```

```
-54.0 114.0
```

In [19]:
```
data4[data4['Duration']>119.0].shape
```

Out[19]:
```
(4884, 18)
```

In [20]:
```
data4.iloc[:,4].value_counts()
```

Out[20]:
```
0.00     35117
7.70      1657
17.82     1546
23.76     1307
6.30      1150
         ...
5.48         1
2.47         1
52.85        1
74.26        1
2.87         1
Name: Commision (in value), Length: 701, dtype: int64
```

In [21]:
```
#comision
Q1=data4.iloc[:,4].quantile(0.25)
Q3=data4.iloc[:,4].quantile(0.75)
LB=Q1-1.5*(Q3-Q1)
UB=Q3+1.5*(Q3-Q1)
print(LB,UB)
```

```
-15.195 25.325000000000003
```

In [22]:
```
data4[data4['Commision (in value)']<-15.195].shape
```

Out[22]:
```
(0, 18)
```

In [23]:
```
data4[data4['Commision (in value)']>25.325000000000003].shape
```

Out[23]:
```
(6462, 18)
```

handle duplicated data

In [24]:
```
data.duplicated().sum()
```

Out[24]:
```
8042
```

In [25]:
```
data1=data.drop_duplicates()
data1
```

Out[25]:

	Agency	Agency Type	Distribution Channel	Product Name	Claim	Duration	Destination	Net Sales	Comm (in
0	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	
1	CBH	Travel Agency	Offline	Comprehensive Plan	No	186	MALAYSIA	-29.0	
2	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	65	AUSTRALIA	-49.5	
3	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	60	AUSTRALIA	-39.6	
4	CWT	Travel Agency	Online	Rental Vehicle Excess Insurance	No	79	ITALY	-19.8	
...	...	...	...	...	...	...	...	...	
63320	JZI	Airlines	Online	Basic Plan	No	5	BRUNEI DARUSSALAM	18.0	
63321	JZI	Airlines	Online	Basic Plan	No	111	JAPAN	35.0	
63322	JZI	Airlines	Online	Basic Plan	No	58	CHINA	40.0	
63323	JZI	Airlines	Online	Basic Plan	No	2	MALAYSIA	18.0	
63325	JZI	Airlines	Online	Basic Plan	No	22	HONG KONG	26.0	

55284 rows × 11 columns

In [26]:
```
data2=data1[(data['Product Name']=='Cancellation Plan') |
           (data['Product Name']=='2 way Comprehensive Plan') |
           (data['Product Name']=='Rental Vehicle Excess Insurance') |
           (data['Product Name']=='Basic Plan') |
           (data['Product Name']=='Bronze Plan') |
           (data['Product Name']=='1 way Comprehensive Plan') |
           (data['Product Name']=='Value Plan') |
           (data['Product Name']=='Silver Plan') |
           (data['Product Name']=='Annual Silver Plan') |
           (data['Product Name']=='Ticket Protector')].reset_index().drop('index'

data3=data2[data2['Age']>18].reset_index().drop('index',axis=1)
from sklearn import preprocessing
data4=data3.drop(['Gender','Agency','Destination'],axis=1)

label_encoder1 = preprocessing.LabelEncoder()
data4['Distribution Channel']= label_encoder1.fit_transform(data4['Distribution C

label_encoder2 = preprocessing.LabelEncoder()
data4['Claim']= label_encoder2.fit_transform(data4['Claim'])

data5=pd.get_dummies(data4)
data5
```

```
<ipython-input-26-93ea0ebb6e10>:1: UserWarning: Boolean Series key will be rein
dexed to match DataFrame index.
  data2=data1[(data['Product Name']=='Cancellation Plan') |
```

Out[26]:

	Distribution Channel	Claim	Duration	Net Sales	Commision (in value)	Age	Agency Type_Airlines	Agency Type_Travel Agency	P Name Compreh
0	1	0	65	-49.5	29.70	32	0	1	
1	1	0	60	-39.6	23.76	32	0	1	
2	1	0	79	-19.8	11.88	41	0	1	
3	1	0	66	-121.0	42.35	44	1	0	
4	1	0	47	-39.6	23.76	32	0	1	
...	...	...	...	...	...	...	...	...	
52580	1	0	5	18.0	6.30	27	1	0	
52581	1	0	111	35.0	12.25	31	1	0	
52582	1	0	58	40.0	14.00	40	1	0	
52583	1	0	2	18.0	6.30	57	1	0	
52584	1	0	22	26.0	9.10	35	1	0	

52585 rows × 18 columns

In [27]:
```
data5.isna().sum()
```

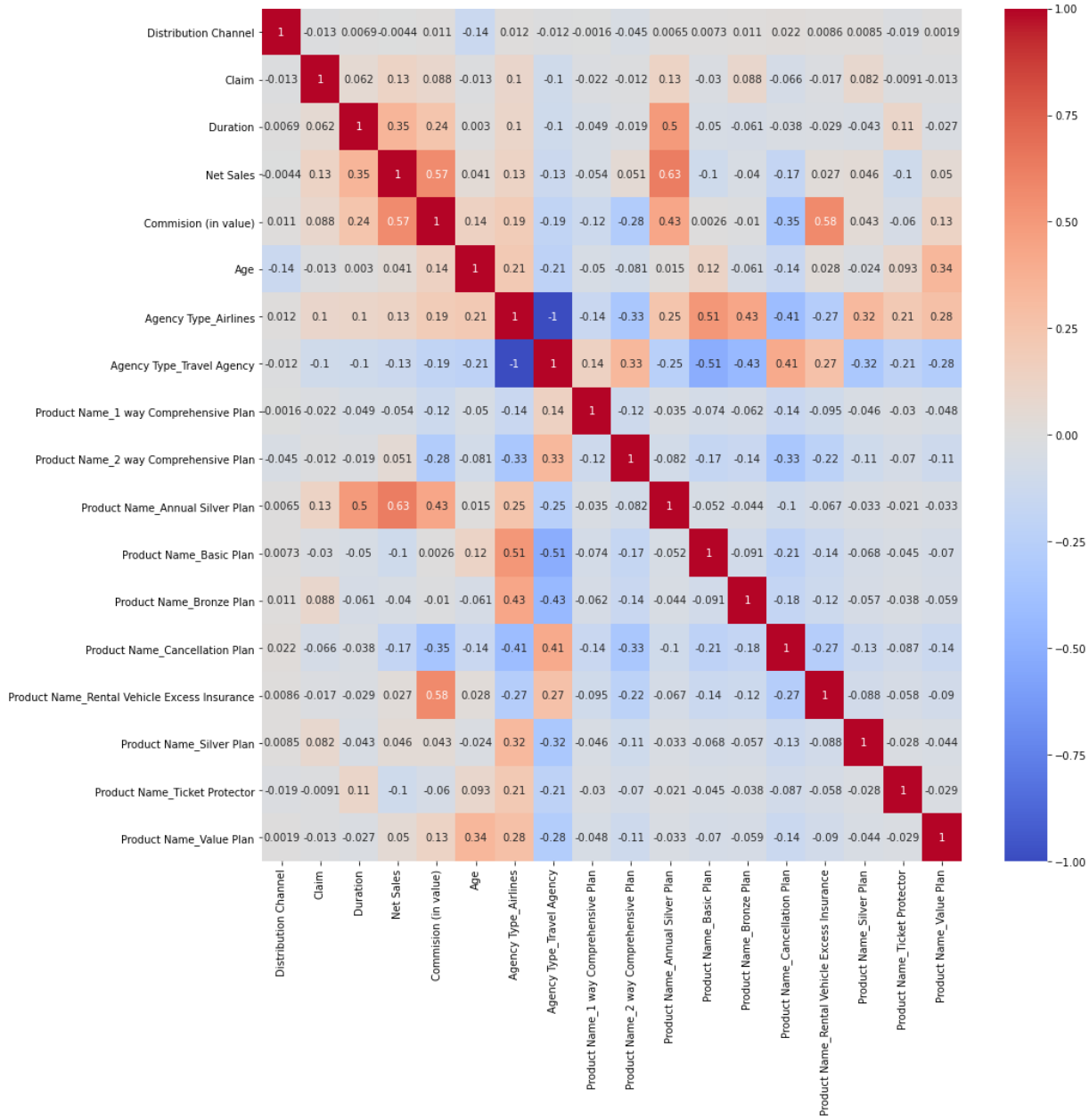
Out[27]:
```
Distribution Channel                            0
Claim                                           0
Duration                                        0
Net Sales                                       0
Commision (in value)                            0
Age                                             0
Agency Type_Airlines                            0
Agency Type_Travel Agency                       0
Product Name_1 way Comprehensive Plan           0
Product Name_2 way Comprehensive Plan           0
Product Name_Annual Silver Plan                 0
Product Name_Basic Plan                         0
Product Name_Bronze Plan                        0
Product Name_Cancellation Plan                  0
Product Name_Rental Vehicle Excess Insurance    0
Product Name_Silver Plan                        0
Product Name_Ticket Protector                   0
Product Name_Value Plan                         0
dtype: int64
```

feature selection

brute force

In [28]:
```
import seaborn as sns
fig=plt.figure(figsize=(15,15))
sns.heatmap(data5.corr(),vmin=-1,vmax=1,annot=True,cmap='coolwarm')
```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x17e00303f40>



filter method

In [29]:
```
data5.shape
```

Out[29]:
```
(52585, 18)
```

In [30]:
```
x=data5.iloc[:,[2,3,4,5]]
binary=data5.iloc[:,[0,6,7,8,9,10,11,12,13,14,15,16,17]].reset_index().drop('inde
y=data5.iloc[:,1]
from sklearn.preprocessing import MinMaxScaler
mm=MinMaxScaler()
x_mm=mm.fit_transform(x)
x_mm
```

Out[30]:
```
array([[0.01372107, 0.28315263, 0.1047619 , 0.13131313],
       [0.01269711, 0.29140951, 0.08380952, 0.13131313],
       [0.01658816, 0.30792327, 0.04190476, 0.22222222],
       ...,
       [0.01228753, 0.35779817, 0.04938272, 0.21212121],
       [0.00081917, 0.33944954, 0.02222222, 0.38383838],
       [0.00491501, 0.34612177, 0.03209877, 0.16161616]])
```

In [31]:
```
mm_df=pd.DataFrame(x_mm,columns=x.columns)
mm_df
```

Out[31]:

	Duration	Net Sales	Commision (in value)	Age
0	0.013721	0.283153	0.104762	0.131313
1	0.012697	0.291410	0.083810	0.131313
2	0.016588	0.307923	0.041905	0.222222
3	0.013926	0.223520	0.149383	0.252525
4	0.010035	0.291410	0.083810	0.131313
...	...	...	...	...
52580	0.001434	0.339450	0.022222	0.080808
52581	0.023142	0.353628	0.043210	0.121212
52582	0.012288	0.357798	0.049383	0.212121
52583	0.000819	0.339450	0.022222	0.383838
52584	0.004915	0.346122	0.032099	0.161616

52585 rows × 4 columns

In [32]:
```
data6=pd.concat([mm_df,binary,y],axis=1)
data6
```

Out[32]:

	Duration	Net Sales	Commision (in value)	Age	Distribution Channel	Agency Type_Airlines	Agency Type_Travel Agency	P Name Compreh
0	0.013721	0.283153	0.104762	0.131313	1	0	1	
1	0.012697	0.291410	0.083810	0.131313	1	0	1	
2	0.016588	0.307923	0.041905	0.222222	1	0	1	
3	0.013926	0.223520	0.149383	0.252525	1	1	0	
4	0.010035	0.291410	0.083810	0.131313	1	0	1	
...	...	...	...	...	...	...	...	
52580	0.001434	0.339450	0.022222	0.080808	1	1	0	
52581	0.023142	0.353628	0.043210	0.121212	1	1	0	
52582	0.012288	0.357798	0.049383	0.212121	1	1	0	
52583	0.000819	0.339450	0.022222	0.383838	1	1	0	
52584	0.004915	0.346122	0.032099	0.161616	1	1	0	

52585 rows × 18 columns

In [33]:
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.feature_selection import f_classif
```

In [34]:
```
x1=data6.iloc[:,:17]
y1=data6.iloc[:,-1]
bestfeatures=SelectKBest(chi2,k=16)
fit=bestfeatures.fit(x1,y1)
dfscores=pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(x1.columns)
featurescores = pd.concat([dfcolumns, dfscores], axis = 1)
featurescores.columns = ["feature_name", "feature_score"]
print(featurescores.nlargest(8, "feature_score"))
```

```
                       feature_name  feature_score
9   Product Name_Annual Silver Plan     868.355257
5              Agency Type_Airlines     382.896789
11         Product Name_Bronze Plan     375.389268
14         Product Name_Silver Plan     337.624184
12   Product Name_Cancellation Plan     163.647460
6         Agency Type_Travel Agency     156.992406
2              Commision (in value)      46.214702
10          Product Name_Basic Plan      42.295518
```

In [35]:
```
x1=data6.iloc[:,:17]
y1=data6.iloc[:,-1]
bestfeatures=SelectKBest(score_func=f_classif,k=16)
fit=bestfeatures.fit(x1,y1)
dfscores=pd.DataFrame(fit.scores_)
dfcolumns=pd.DataFrame(x1.columns)
featuresscores=pd.concat([dfcolumns,dfscores],axis=1)
featuresscores.columns=['feature_name','feature_score']
print(featurescores.nlargest(8,'feature_score'))
```

```
                       feature_name  feature_score
9   Product Name_Annual Silver Plan     868.355257
5              Agency Type_Airlines     382.896789
11         Product Name_Bronze Plan     375.389268
14         Product Name_Silver Plan     337.624184
12   Product Name_Cancellation Plan     163.647460
6         Agency Type_Travel Agency     156.992406
2              Commision (in value)      46.214702
10          Product Name_Basic Plan      42.295518
```

embeded

handle imbalance data

In [36]:
```
data6.Claim.value_counts()
```

Out[36]:
```
0    51757
1      828
Name: Claim, dtype: int64
```

In [37]:
```
class_0=data6[data6['Claim']==0]
class_1=data6[data6['Claim']==1]
```

In [38]:
```
class_count_0, class_count_1 = data.Claim.value_counts()
```

In [39]:
```
class1over=class_1.sample(class_count_0,replace=True)
class1over
```

Out[39]:

	Duration	Net Sales	Commision (in value)	Age	Distribution Channel	Agency Type_Airlines	Agency Type_Travel Agency	P Name Compreh
46880	0.075364	0.504587	0.190476	0.232323	1	1	0	
40568	0.074954	0.504587	0.190476	0.111111	1	1	0	
4170	0.080279	0.575480	0.265432	0.292929	1	1	0	
45413	0.004096	0.351960	0.000000	1.000000	0	0	1	
29107	0.002253	0.344037	0.020741	0.101010	1	1	0	
...	...	...	...	...	...	...	...	
50938	0.003277	0.349041	0.026032	0.181818	1	1	0	
9311	0.002048	0.357465	0.083810	0.383838	1	0	1	
2755	0.003072	0.388073	0.067266	0.161616	1	1	0	
23614	0.075159	0.535321	0.222963	0.131313	1	1	0	
26415	0.006553	0.385321	0.000000	0.171717	1	0	1	

62399 rows × 18 columns

In [40]:
```
dataover=pd.concat([class_0,class1over],axis=0).reset_index().drop('index',axis=1
dataover
```

Out[40]:

	Duration	Net Sales	Commision (in value)	Age	Distribution Channel	Agency Type_Airlines	Agency Type_Travel Agency	Nam Compre
0	0.013721	0.283153	0.104762	0.131313	1	0	1	
1	0.012697	0.291410	0.083810	0.131313	1	0	1	
2	0.016588	0.307923	0.041905	0.222222	1	0	1	
3	0.013926	0.223520	0.149383	0.252525	1	1	0	
4	0.010035	0.291410	0.083810	0.131313	1	0	1	
...	...	...	...	...	...	...	...	
114151	0.003277	0.349041	0.026032	0.181818	1	1	0	
114152	0.002048	0.357465	0.083810	0.383838	1	0	1	
114153	0.003072	0.388073	0.067266	0.161616	1	1	0	
114154	0.075159	0.535321	0.222963	0.131313	1	1	0	
114155	0.006553	0.385321	0.000000	0.171717	1	0	1	

114156 rows × 18 columns

splitting dataset into the training set and test set

In [41]:
```
X=dataover.iloc[:,:-1]
Y=dataover.iloc[:,-1]
```

In [42]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.3,random_state=0
```

In [43]:
```
X_train
```

Out[43]:

	Duration	Net Sales	Commision (in value)	Age	Distribution Channel	Agency Type_Airlines	Agency Type_Travel Agency	Nam Compre
20471	0.001434	0.349208	0.062857	0.161616	1	0	1	
97913	0.002048	0.346956	0.023810	0.070707	1	1	0	
37476	0.024985	0.341952	0.018519	0.040404	1	1	0	
83540	0.026828	0.364470	0.000000	0.171717	1	0	1	
103438	0.003686	0.359466	0.000000	0.171717	1	0	1	
...	...	...	...	...	...	...	...	
21243	0.028057	0.332777	0.000000	0.020202	1	0	1	
45891	0.003891	0.332694	0.020952	0.323232	1	0	1	
42613	0.010854	0.373978	0.125714	0.404040	1	0	1	
43567	0.001638	0.337781	0.014109	0.181818	1	1	0	
68268	0.015359	0.464804	0.356190	0.525253	1	0	1	

79909 rows × 17 columns

In [44]:
```
X_test
```

Out[44]:

	Duration	Net Sales	Commision (in value)	Age	Distribution Channel	Agency Type_Airlines	Agency Type_Travel Agency	Nam Compre
29978	0.004096	0.324437	0.062857	0.050505	1	0	1	
13529	0.010444	0.382235	0.146667	0.131313	1	0	1	
103154	0.006963	0.334445	0.000000	0.161616	1	0	1	
68608	0.003277	0.352377	0.029559	0.131313	1	1	0	
79945	0.001638	0.385321	0.000000	0.171717	1	0	1	
...	...	...	...	...	...	...	...	
87629	0.075773	0.535321	0.222963	0.101010	1	1	0	
30999	0.002867	0.382235	0.146667	0.121212	1	0	1	
8406	0.001434	0.339450	0.022222	0.323232	1	1	0	
82371	0.013516	0.383653	0.062610	0.141414	1	1	0	
56922	0.002253	0.341118	0.017637	0.020202	1	1	0	

34247 rows × 17 columns

In [45]:
```
Y_train
```

Out[45]:
```
20471     0
97913     1
37476     0
83540     1
103438    1
         ..
21243     0
45891     0
42613     0
43567     0
68268     1
Name: Claim, Length: 79909, dtype: int32
```

In [46]:
```
Y_test
```

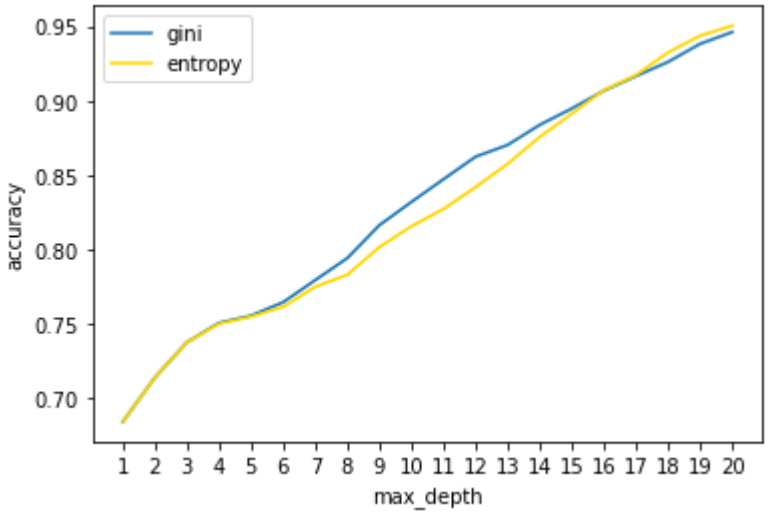
Out[46]:
```
29978     0
13529     0
103154    1
68608     1
79945     1
         ..
87629     1
30999     0
8406      0
82371     1
56922     1
Name: Claim, Length: 34247, dtype: int32
```

traing the decision tree model on training test

In [47]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
max_depth=[]
acc_gini=[]
acc_entropy=[]
for i in range(1,21):
    dtree=DecisionTreeClassifier(criterion='gini',max_depth=i)
    dtree.fit(X_train,Y_train)
    y_pred=dtree.predict(X_test)
    acc_gini.append(accuracy_score(Y_test,y_pred))

    dtree=DecisionTreeClassifier(criterion='entropy',max_depth=i)
    dtree.fit(X_train,Y_train)
    y_pred=dtree.predict(X_test)
    acc_entropy.append(accuracy_score(Y_test,y_pred))

    max_depth.append(i)
df=pd.DataFrame({'acc_gini':pd.Series(acc_gini),'acc_entropy':pd.Series(acc_entro
plt.plot('max_depth','acc_gini',data=df,label='gini')
plt.plot('max_depth','acc_entropy',data=df,label='entropy',color='gold')
plt.xlabel('max_depth')
plt.ylabel('accuracy')
plt.xticks([j for j in range(1,21)])
plt.legend()
plt.show()
```



building model

In [48]:
```
classifier=DecisionTreeClassifier(criterion='gini',max_depth=20)
classifier.fit(X_train,Y_train)
```

Out[48]:
```
DecisionTreeClassifier(max_depth=20)
```

visualization

In [49]:
```
!pip install --upgrade scikit-learn==0.20.3
```

```
  atlas_3_10_blas_info:
    libraries satlas not found in ['C:\\Users\\sun\\anaconda3\\lib', 'C:\\',
'C:\\Users\\sun\\anaconda3\\libs']
    NOT AVAILABLE

  atlas_blas_threads_info:
  Setting PTATLAS=ATLAS
    libraries ptf77blas,ptcblas,atlas not found in ['C:\\Users\\sun\\anaconda
3\\lib', 'C:\\', 'C:\\Users\\sun\\anaconda3\\libs']
    NOT AVAILABLE

  atlas_blas_info:
    libraries f77blas,cblas,atlas not found in ['C:\\Users\\sun\\anaconda3\\l
ib', 'C:\\', 'C:\\Users\\sun\\anaconda3\\libs']
    NOT AVAILABLE

  accelerate_info:
    NOT AVAILABLE


  C:\Users\sun\anaconda3\lib\site-packages\numpy\distutils\system_info.py:189
```

In [50]:
```
!pip install pydotplus
```

```
  copying sklearn\cross_decomposition\pls_.py -> build\lib.win-amd64-3.8\skle
arn\cross_decomposition
  copying sklearn\cross_decomposition\__init__.py -> build\lib.win-amd64-3.8
\sklearn\cross_decomposition
  creating build\lib.win-amd64-3.8\sklearn\cross_decomposition\tests
  copying sklearn\cross_decomposition\tests\test_pls.py -> build\lib.win-amd6
4-3.8\sklearn\cross_decomposition/tests
  copying sklearn\cross_decomposition\tests\__init__.py -> build\lib.win-amd6
4-3.8\sklearn\cross_decomposition/tests
  creating build\lib.win-amd64-3.8\sklearn\feature_selection
  copying sklearn\feature_selection\base.py -> build\lib.win-amd64-3.8\sklear
n\feature_selection
  copying sklearn\feature_selection\from_model.py -> build\lib.win-amd64-3.8
\sklearn\feature_selection
  copying sklearn\feature_selection\mutual_info_.py -> build\lib.win-amd64-3.
8\sklearn\feature_selection
  copying sklearn\feature_selection\rfe.py -> build\lib.win-amd64-3.8\sklearn
\feature_selection
  copying sklearn\feature_selection\univariate_selection.py -> build\lib.win-
```

In [51]:
```
!pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\sun\anaconda3\lib\site-pack
ages (0.17)
```

In [52]:
```
from six import StringIO
from sklearn.tree import export_graphviz
import pydotplus
from IPython.display import Image
dot_data = StringIO()
export_graphviz(classifier, out_file = dot_data, filled = True, precision = 2)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

```
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.612685 to fit
```

Out[52]:

validation

confusion matrix

In [53]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(Y_test,y_pred)
```

Out[53]:
```
array([[14058,  1418],
       [  279, 18492]], dtype=int64)
```

accuracy score

In [54]:
```
accuracy_score(Y_test,y_pred)
```

Out[54]:
```
0.9504482144421409
```

precision and recall

In [55]:
```
from sklearn.metrics import precision_score
precision_score(Y_test, y_pred)
```

Out[55]:
```
0.9287795077850326
```

recall

In [56]:
```
from sklearn.metrics import recall_score
recall_score(Y_test, y_pred)
```

Out[56]:
```
0.9851366469554099
```

F_score

In [57]:
```
from sklearn.metrics import f1_score
f1_score(Y_test, y_pred)
```

Out[57]: 0.9561283317390966

k_fold cross validation

In [58]:
```
from sklearn.model_selection import cross_val_score
estimator = cross_val_score(estimator = classifier, X = X_train, y = Y_train, cv
estimator.mean()
```

Out[58]: 0.9514070209901977