**E-retail factors for customer activation and retention:**

\*A case study from Indian e-commerce customers

**\***Customer satisfaction has emerged **as** one of the most important factors that

guarantee the success of online store**.**

**\***A comprehensive review of the literature, theories **and** models have been

carried out to propose the models **for** customer activation **and** customer retention**.**

**\***Five major factors that contributed to the success of an e**-**commerce store have been

identified **as**: service quality, system quality, information quality, trust **and** net benefit**.**

**\***The research furthermore investigated the factors that influence the online customers

repeat purchase intention**.** The combination of both utilitarian value **and** hedonistic values

are needed to affect the repeat purchase intention (loyalty) positively**.**

**\***The data **is** collected **from** the Indian online shoppers**.**

**\***Results indicate the e**-**retail success factors, which are very much critical **for** customer satisfaction**.**

*#****Importing the libraries***

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

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

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

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

**import** warnings

warnings**.**filterwarnings('ignore')

**!pip install openpyxl**

Requirement already satisfied: openpyxl in c:\users\satya\anaconda3\lib\site-packages (3.0.9)

Requirement already satisfied: et-xmlfile in c:\users\satya\anaconda3\lib\site-packages (from openpyxl) (1.1.0)

*# Reading the data*

Dt**=**pd**.**read\_excel('D:\Data Trained - Excel links\customer\_retention\_dataset.xlsx')

Dt**.**head()

***# Now Setting option to show max rows and max columns***

pd**.**set\_option("display.max\_columns",**None**)

pd**.**set\_option("display.max\_rows", **None**)

In [7]:

**from** string **import** digits

*#Removing tab spaces*

Dt**.**columns **=** Dt**.**columns**.**str**.**replace('\t','')

**EXPLORATORY DATA ANALYTICS :**

*# looking at the sahpe of the data*

Dt**.**shape

(269, 71)

*# looking at all data types*

Data**.**dtypes

*#Removing digits*

remove\_digits **=** str**.**maketrans('', '', digits)

Dt**.**columns **=** Dt**.**columns**.**str**.**translate(remove\_digits)

*#Removing leading and trailling spaces*

Dt**.**columns **=** Dt**.**columns**.**str**.**strip()

*# looking at the head of the data*

Dt**.**head()

*# Let see the null count for each column, but will not count*

Dt**.**isnull()**.**sum()**.**any()

False

*# Let see the the uniqueness of the data*

Dt**.**nunique(

**IMPLEMENTING UNIVARIATE ANALYSIS**

In [15]:

personal\_info**=**['Gender of respondent','How old are you?','Which city do you shop online from?',

'What is the Pin Code of where you shop online from?','Since How Long You are Shopping Online ?',

'How many times you have made an online purchase in the past year?']

**Checking Personal information**

**for** i **in** personal\_info:

**if** i**!=**'What is the Pin Code of where you shop online from?':

plt**.**figure(figsize**=**(8,6))

Dt[i]**.**value\_counts()**.**plot**.**pie(autopct**=**'%1.1f%%')

centre**=**plt**.**Circle((0,0),0.7,fc**=**'white')

fig**=**plt**.**gcf()

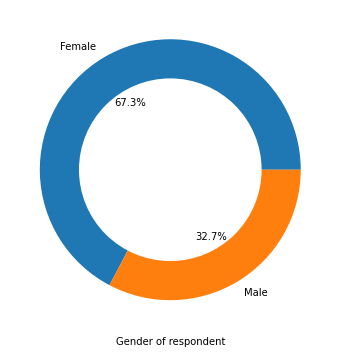
fig**.**gca()**.**add\_artist(centre)

plt**.**xlabel(i)

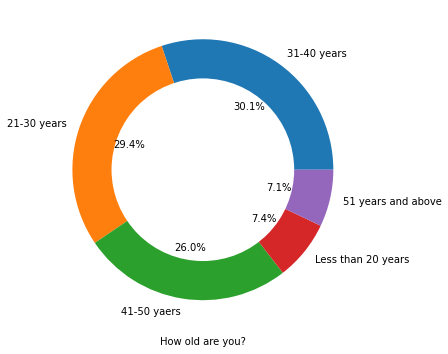
plt**.**ylabel('')

plt**.**figure()

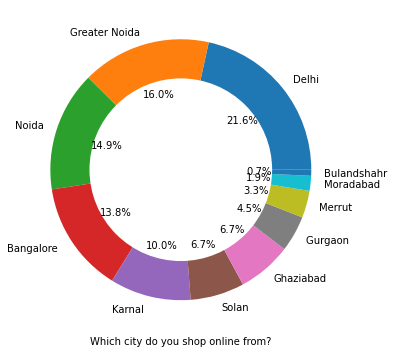
plt**.**figure()



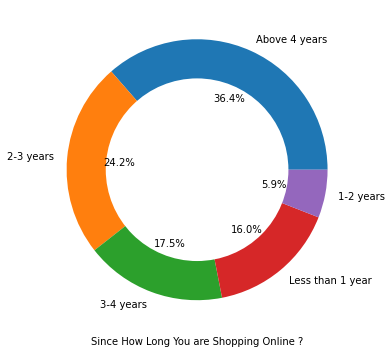
<Figure size 432x288 with 0 Axes>



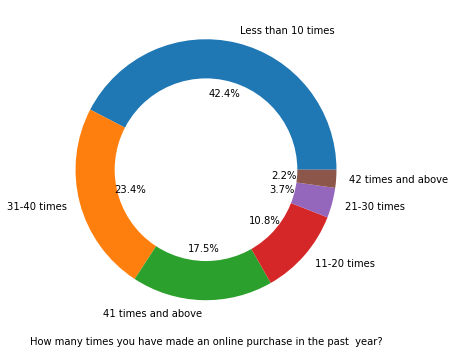
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

Here **in** the analysis, there are double the number of women than men who took the survey**.**

**-**Most of the people are **in** the 30's followed by 20's,senior citizen **and** teenagers are the least **in** number**.**

**-**Most of the people belongs to Banglore, delhi, noida ambiguity can also be seen **as** noida has two categories(noida **and** greater noida) which need to be handled**.**

**-**Most of the people shopping online been shopping **from** a long time**.**

**-**Majority of people shop online 10 times a year, ambiguity can also be seen **for** range 42 times **and** above which needs are to be handeled**.**

Analysing on the basis of Various following factors

**Intention of Repeat purchase:**

*#Resolving ambiguity of column*

*#Changing 42 times and above to 41 times and above*

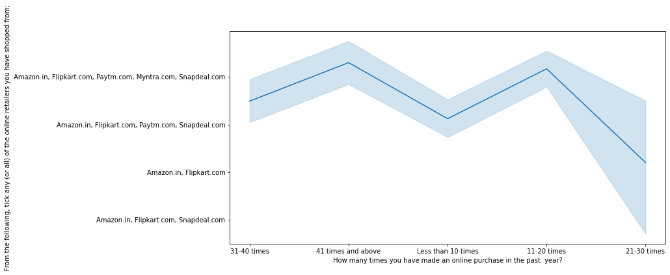
Dt['How many times you have made an online purchase in the past year?']**.**replace('42 times and above','41 times and above',

inplace**=True**)

plt**.**figure(figsize**=**(12,6))

sns**.**lineplot(Dt['How many times you have made an online purchase in the past year?'],

Dt['From the following, tick any (or all) of the online retailers you have shopped from;'])



Most of the shoppers who shop more than 41 times a year shop from all the online brands, some of the people who shop for 32-40 and less than 10 times a year seem to exclude myntra. People shop from flipkart and Amazon whatever be the case.

**\*Now doing Convertion of years to numbers for better analysis**

dict**=**{'31-40 times':35,'41 times and above':45,'Less than 10 times':5,'11-20 times':15,'21-30 times':25}

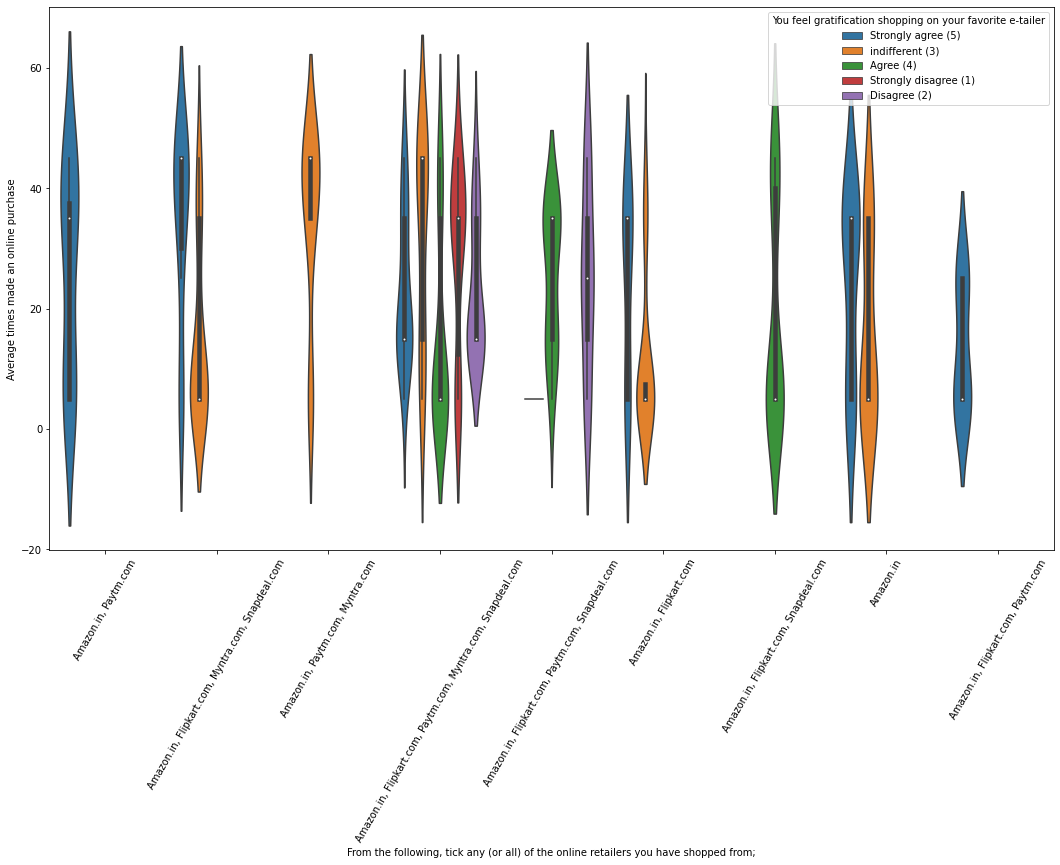
Dt['Average times made an online purchase']**=**Dt['How many times you have made an online purchase in the past year?']**.**replace(dict)

plt**.**figure(figsize**=**(18,10))

sns**.**violinplot(Dt['From the following, tick any (or all) of the online retailers you have shopped from;'],

Dt['Average times made an online purchase'],hue**=**Dt['You feel gratification shopping on your favorite e-tailer'])

plt**.**xticks(rotation**=**60)



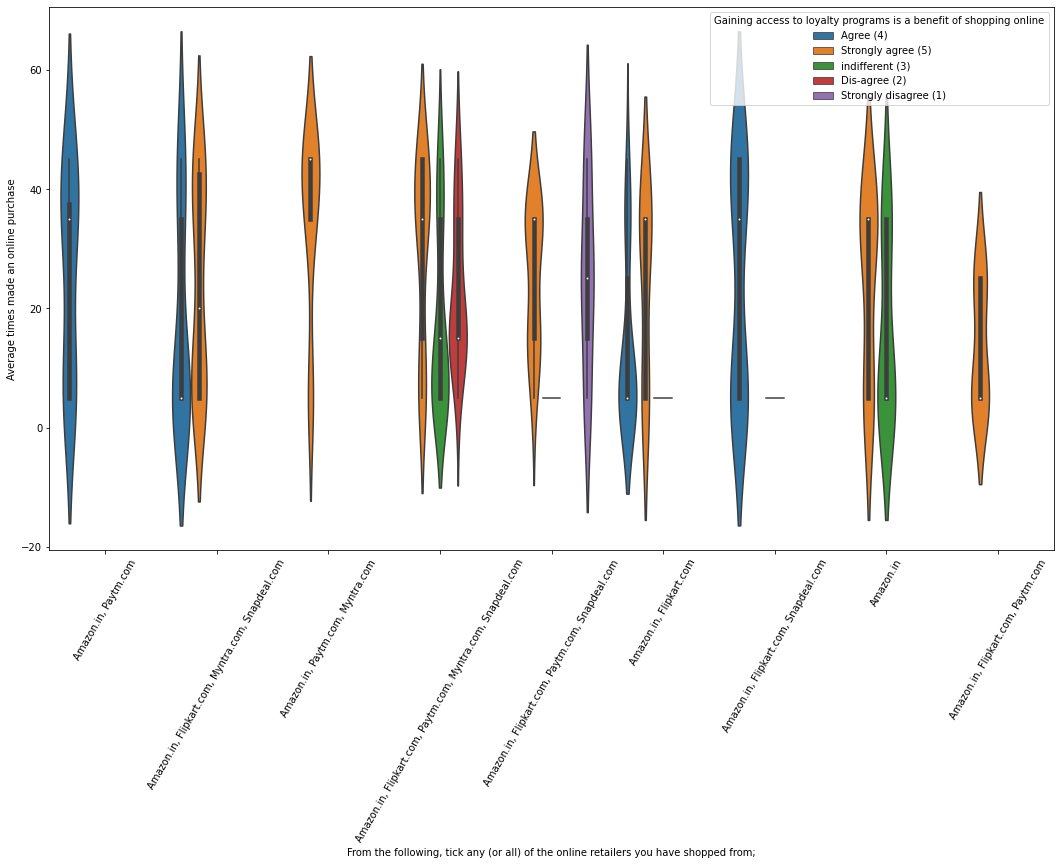
From the graph analysis,All the people who have shopped from amazon,paytm, flipkart are satisfied. People who shop from a more number of online brands dosnot get satisfied.

plt**.**figure(figsize**=**(18,10))

sns**.**violinplot(Dt['From the following, tick any (or all) of the online retailers you have shopped from;'],

Dt['Average times made an online purchase'],hue**=**Dt['Gaining access to loyalty programs is a benefit of shopping online'])

plt**.**xticks(rotation**=**60)

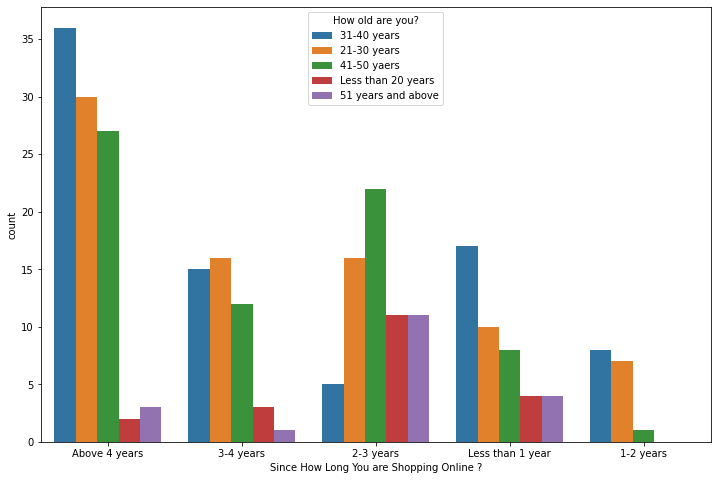


From the Grapgh analysis, people shopping from flipkart and sanpdeal seems to give such benefits but people who shop from almost everywhere disagree with this statement too. Moreover, People of Amazon and paytm are getting benefits from the loyalty points

**Getting Online Retailing Analysis:**

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

sns**.**countplot(Dt['Since How Long You are Shopping Online ?'],hue**=**Dt['How old are you?'])



More and Highest number of people have been shopping online for above 4 years except for the age group below 20 years and above 50 years.People who are shopping online for 1-2 years does not include teenagers and elder people.

**#Converting Years to numbers for better analysis**

In [31]:

*# Getting data about the age groups*

Dt['Since How Long You are Shopping Online ?']**.**unique()

Out[31]:

array(['Above 4 years', '3-4 years', '2-3 years', 'Less than 1 year',

'1-2 years'], dtype=object)

In [32]:

dict**=**{'Above 4 years':4.5,'3-4 years':3.5,'2-3 years':2.5,'1-2 years':1.5,'Less than 1 year':0.5}

Dt['Average years of shopping online']**=**Dt['Since How Long You are Shopping Online ?']**.**replace(dict)

In [34]:

*# Getting data about the cities*

Dt['Which city do you shop online from?']**.**unique()

Out[34]:

array(['Delhi', 'Greater Noida', 'Karnal ', 'Bangalore ', 'Noida',

'Solan', 'Moradabad', 'Gurgaon ', 'Merrut', 'Ghaziabad',

'Bulandshahr'], dtype=object)

*#Changing Greater noida to noida*

Dt['Which city do you shop online from?']**.**replace({'Greater Noida':'Noida'},inplace**=True**)

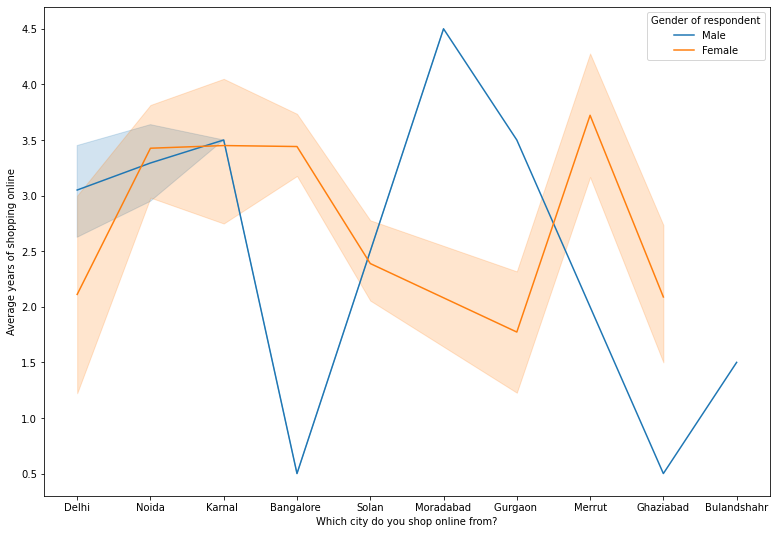
In [37]:

plt**.**figure(figsize**=**(13,9))

sns**.**lineplot(Dt['Which city do you shop online from?'],Dt['Average years of shopping online'],hue**=**Dt['Gender of respondent'])

Out[37]:

<AxesSubplot:xlabel='Which city do you shop online from?', ylabel='Average years of shopping online'>



In the above given data lines,one can say that, the density of female customers is more than male. *Highest number of men shopping online belong from delhi and noida,*whereas, men from moradabad have been shopping online for the longest. *Men living in banglore and ghaziabad shop have shopped online for less than 1 year.*Women from meerut and noida have shopped the longest.

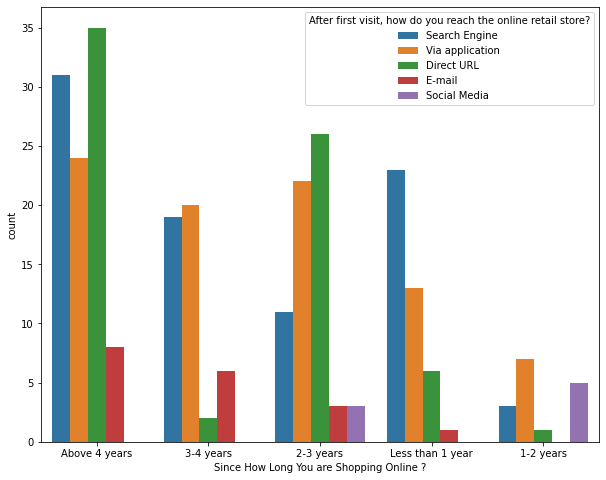
In [38]:

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

sns**.**countplot(Dt['Since How Long You are Shopping Online ?'],

hue**=**Dt['After first visit, how do you reach the online retail store?'])

<AxesSubplot:xlabel='Since How Long You are Shopping Online ?', ylabel='count'>



Moreover, people who are shopping online for more than 3 years donot use the application rather use search engine and direct url's in large number which indicates that online brands should update all their platforms rather than just application.

# Checking for Brand image

performance**=**['Easy to use website or application',

'Visual appealing web-page layout', 'Wild variety of product on offer',

'Complete, relevant description information of products',

'Fast loading website speed of website and application',

'Reliability of the website or application',

'Quickness to complete purchase',

'Availability of several payment options', 'Speedy order delivery',

'Privacy of customers’ information',

'Security of customer financial information',

'Perceived Trustworthiness',

'Presence of online assistance through multi-channel']

In [41]:

**for** i **in** performance:

plt**.**figure(figsize**=**(9,7))

Dt[i]**.**value\_counts()**.**plot**.**pie(autopct**=**'%1.1f%%')

centre**=**plt**.**Circle((0,0),0.7,fc**=**'yellow')

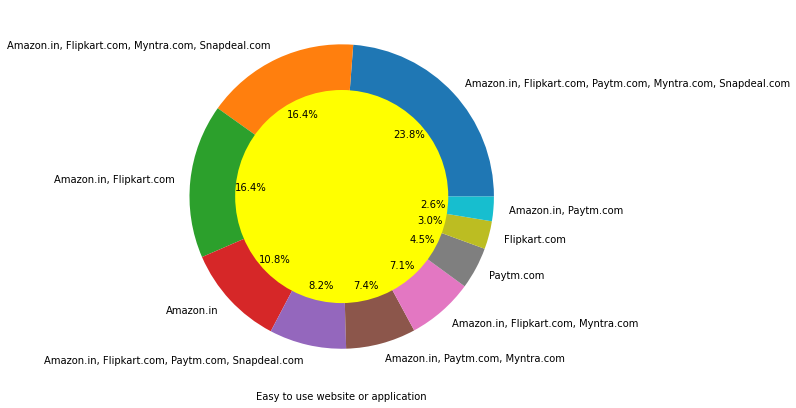
fig**=**plt**.**gcf()

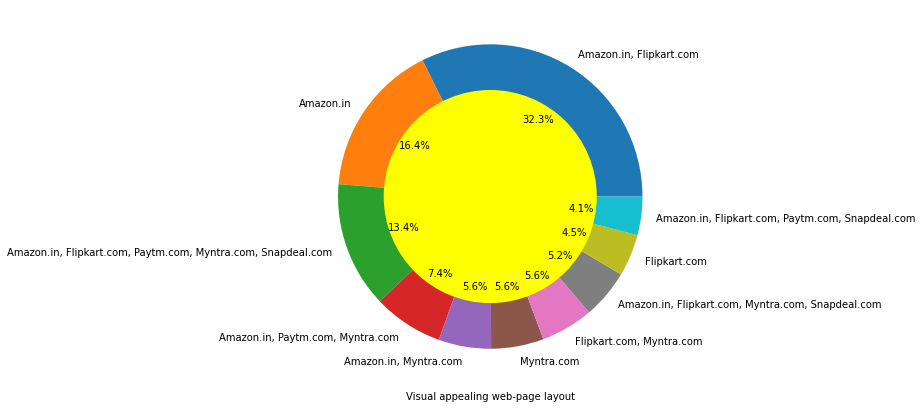
fig**.**gca()**.**add\_artist(centre)

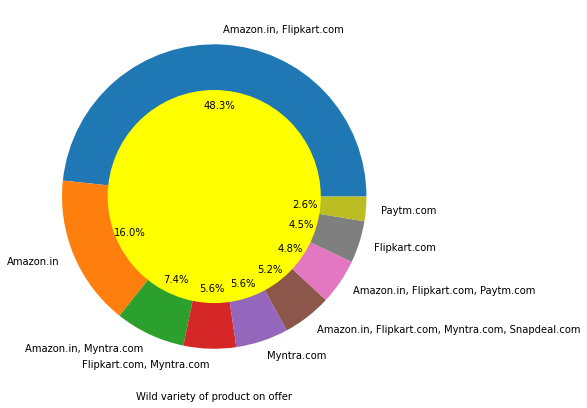
plt**.**xlabel(i)

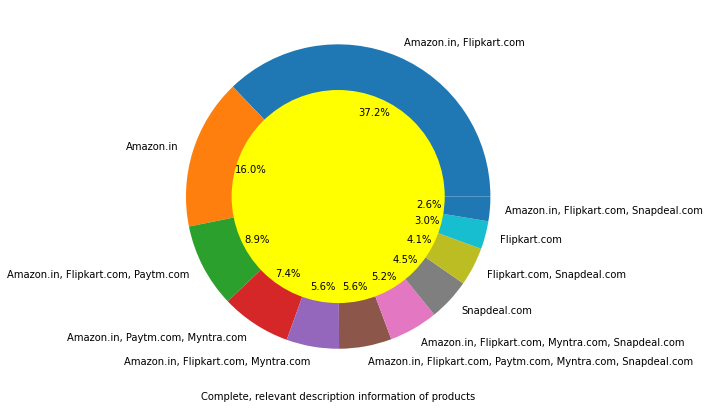
plt**.**ylabel('')

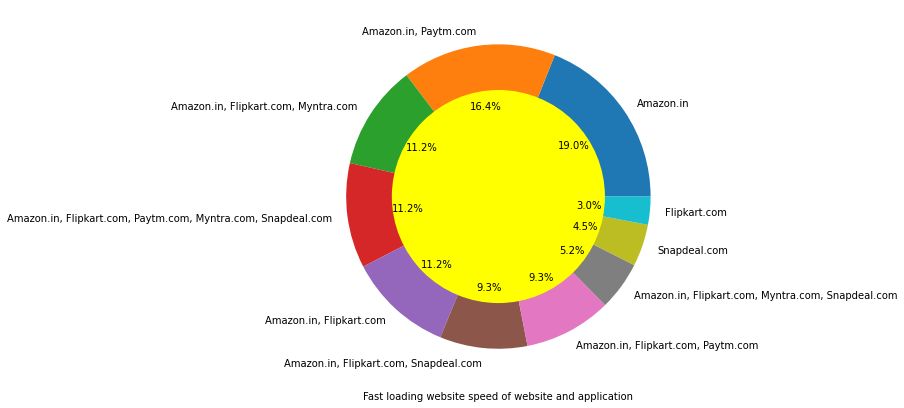
plt**.**figure()

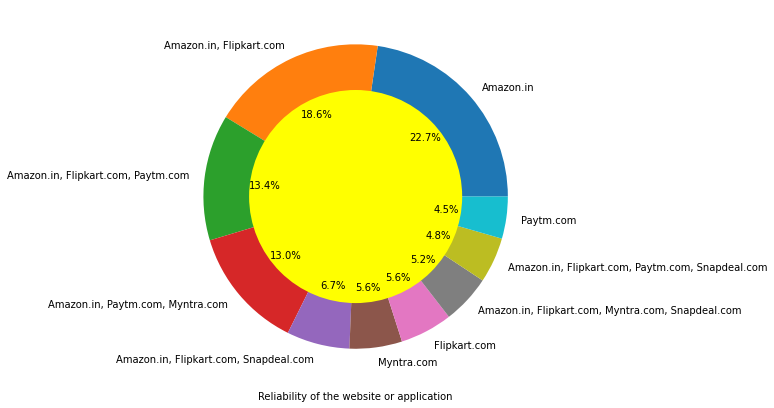


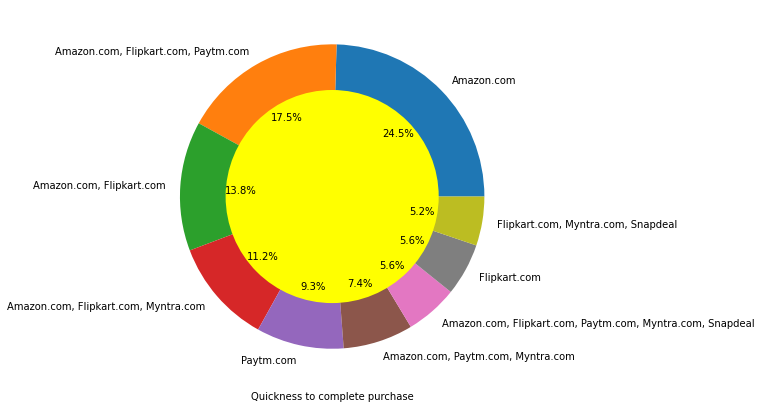












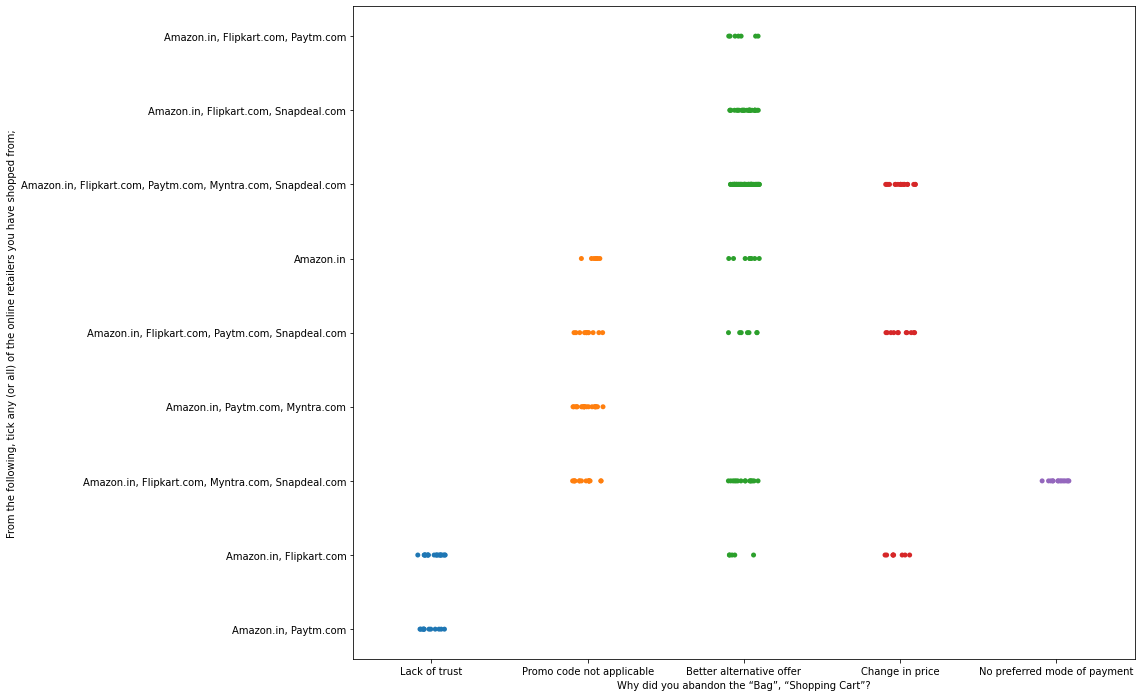
Flipkart,Amazon have been the highest votes for having all the positive points and have maintained a very good brand image followed by paytm and the myntra.

In [42]:

plt**.**figure(figsize**=**(14,12))

sns**.**stripplot(Dt['Why did you abandon the “Bag”, “Shopping Cart”?'],

Dt['From the following, tick any (or all) of the online retailers you have shopped from;'])



*We can clearly see that most of the time people avoid the bag is beacuse they get a better alternative offer or promo code not applicable.*There is also lack of trust seen in flipkart,amazon and paytm by some people.

**Checking Loyalty**

The people who are Loyal customers keep using the same brand even if it is not good as other brands

In [43]:

*#Collecting all the negative remarks about a brand*

bad**=**['Longer time to get logged in (promotion, sales period)',

'Longer time in displaying graphics and photos (promotion, sales period)',

'Late declaration of price (promotion, sales period)',

'Longer page loading time (promotion, sales period)',

'Limited mode of payment on most products (promotion, sales period)',

'Longer delivery period', 'Change in website/Application design',

'Frequent disruption when moving from one page to another']

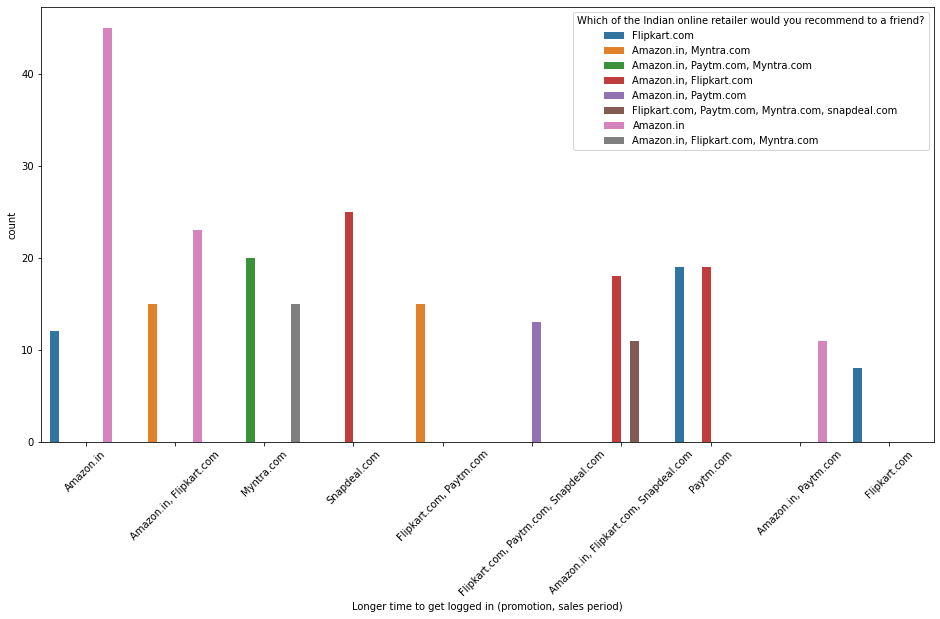
**for** i **in** bad:

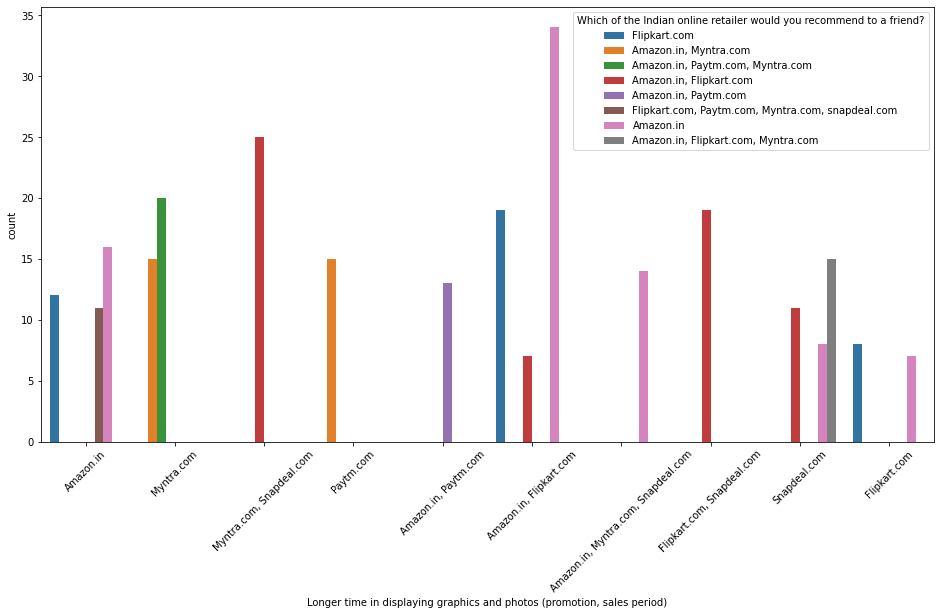
plt**.**figure(figsize**=**(16,8))

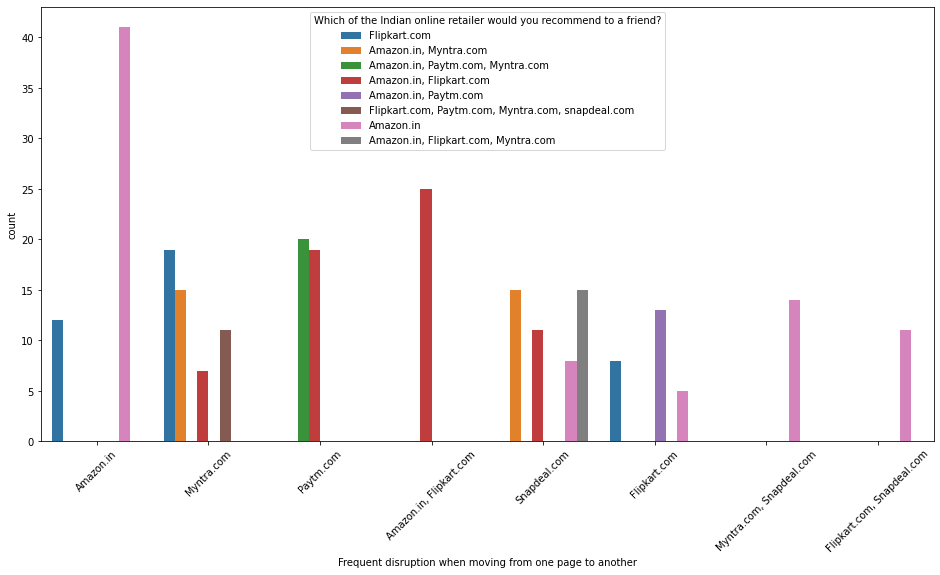
sns**.**countplot(Dt[i],hue**=**Dt['Which of the Indian online retailer would you recommend to a friend?'])

plt**.**xticks(rotation**=**45)

plt**.**figure()







<Figure size 432x288 with 0 Axes>

\*Customers seem to be more loyal to amazon, flipkart and paytm as even though many of them have given negative remarks about them still they would recommend these platforms to their friend.

**Processing the dataframe**

\*Separating the label from rest of the features

In [45]:

x**=**Dt**.**copy()

x**.**drop('Which of the Indian online retailer would you recommend to a friend?',axis**=**1,inplace**=True**)

y**=**Dt['Which of the Indian online retailer would you recommend to a friend?']

**Encoding Categorical Features**

In [46]:

cat**=**[i **for** i **in** x**.**columns **if** x[i]**.**dtypes**==**'O']

In [47]:

*# Getting the libraries*

**from** sklearn.preprocessing **import** OrdinalEncoder,LabelEncoder

encode**=**OrdinalEncoder()

labe**=**LabelEncoder()

*#using ordinal encoder for independent features*

**for** i **in** cat:

x[i]**=**encode**.**fit\_transform(x[i]**.**values**.**reshape(**-**1,1))

*#Using label encoder for Label Column*

y**=**labe**.**fit\_transform(y)

**Doing Scaling Process:**

In [49]:

**from** sklearn.preprocessing **import** MinMaxScaler

scaler**=**MinMaxScaler()

xd**=**scaler**.**fit\_transform(x)

x**=**pd**.**DataFrame(xd,columns**=**x**.**columns)

**Using chi2 test**

**from** sklearn.feature\_selection **import** SelectKBest

**from** sklearn.feature\_selection **import** chi2

In [59]:

selection **=** SelectKBest(score\_func**=**chi2)

fit **=** selection**.**fit(x,y)

In [60]:

*#Naming the dataframe columns*

Datascores **=** pd**.**DataFrame(fit**.**scores\_)

Datacolumns **=** pd**.**DataFrame(x**.**columns)

featureScores **=** pd**.**concat([Datacolumns,Datascores],axis**=**1)

featureScores**.**columns **=** ['Features','Score']

In [61]:

*# print 10 best features*

print(featureScores**.**nlargest(10,'Score'))

feat**=**list(featureScores**.**nlargest(10,'Score')['Features'])

Features Score

16 Why did you abandon the “Bag”, “Shopping Cart”? 75.754028

22 Loading and processing speed 59.810983

42 Shopping on the website gives you the sense of... 59.253569

10 What browser do you run on your device to acce... 57.171099

67 Change in website/Application design 55.301526

49 Visual appealing web-page layout 54.245760

65 Limited mode of payment on most products (prom... 53.269266

61 Longer time to get logged in (promotion, sales... 48.222655

62 Longer time in displaying graphics and photos ... 48.130643

50 Wild variety of product on offer 47.605973

Using various feature selection method which affects the most

**#Using Feature importance of random forrest**

In [51]:

**from** sklearn.ensemble **import** RandomForestClassifier

m**=**RandomForestClassifier()

m**.**fit(x,y)

andomForestClassifier()

In [52]:

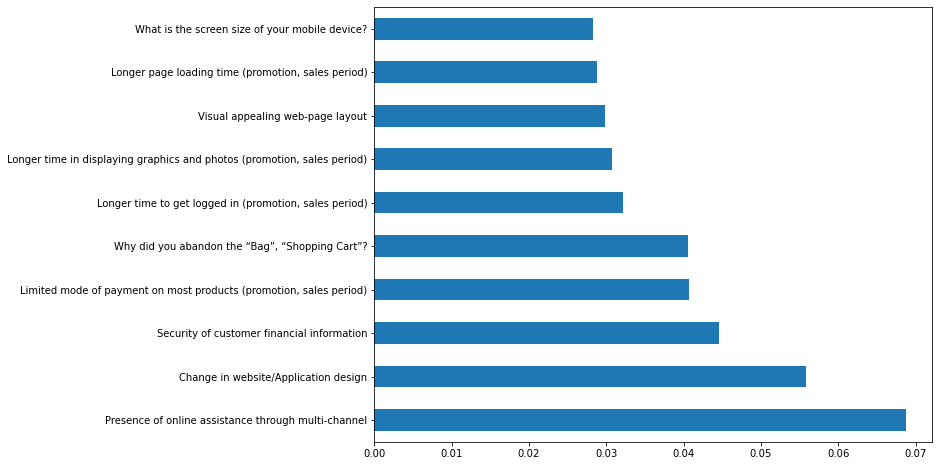
*# Now plot graph of feature importances for better visualization*

feat\_importances **=** pd**.**Series(m**.**feature\_importances\_, index**=**x**.**columns)

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

feat\_importances**.**nlargest(10)**.**plot(kind**=**'barh')

plt**.**show()



From the above chart we can see that above features are of most importance in determining whhich platform will a ciustomer recommend to his friend.

**#PCA Anaysis**

In [62]:

**from** sklearn.decomposition **import** PCA

pca **=** PCA()**.**fit(x)

In [63]:

fig, ax **=** plt**.**subplots(figsize**=**(20,10))

xi **=** np**.**arange(1, 73, step**=**1)

yi **=** np**.**cumsum(pca**.**explained\_variance\_ratio\_)

plt**.**ylim(0.0,1.1)

plt**.**plot(xi, yi, marker**=**'o', linestyle**=**'--', color**=**'b')

plt**.**xlabel('Number of Components')

plt**.**xticks(np**.**arange(0, 72, step**=**1)) *#change from 0-based array index to 1-based human-readable label*

plt**.**ylabel('Cumulative variance (%)')

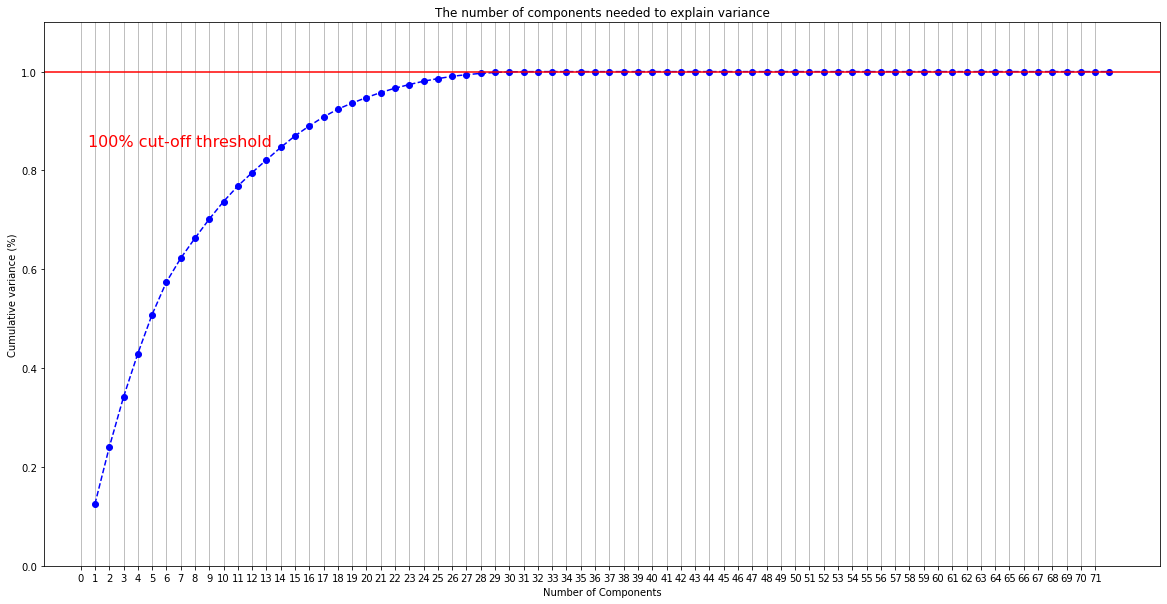
plt**.**title('The number of components needed to explain variance')

plt**.**axhline(y**=**1, color**=**'r', linestyle**=**'-')

plt**.**text(0.5, 0.85, '100% cut-off threshold', color **=** 'red', fontsize**=**16)

ax**.**grid(axis**=**'x')

plt**.**show()



* The grapgh shows number of components are increased from 0.1 to 1.0 and growing constantly

In [64]:

pca**=**PCA(n\_components**=**29)

x**=**pca**.**fit\_transform(x)

x**=**pd**.**DataFrame(x)

x**.**head()

* From the Table- the values what we get are both nagative and positve results

**Modelling Phase**

In [66]:

**from** sklearn.model\_selection **import** train\_test\_split,cross\_val\_score

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.metrics **import** accuracy\_score,confusion\_matrix,classification\_report,roc\_auc\_score,roc\_curve

In [67]:

xtrain,xtest,ytrain,ytest**=**train\_test\_split(x,y,test\_size**=**0.3,random\_state**=**7)

**Random Forest Analysis**

model**=**RandomForestClassifier()

model**.**fit(xtrain,ytrain)

p**=**model**.**predict(xtest)

s**=**cross\_val\_score(model,x,y,cv**=**10)

print('Accuracy',np**.**round(accuracy\_score(p,ytest),4))

print('----------------------------------------------------------')

print('Mean of Cross Validation Score',np**.**round(s**.**mean(),4))

print('----------------------------------------------------------')

print('Confusion Matrix')

print(confusion\_matrix(p,ytest))

print('----------------------------------------------------------')

print('Classification Report')

print(classification\_report(p,ytest))

Accuracy 1.0

----------------------------------------------------------

Mean of Cross Validation Score 0.9889

----------------------------------------------------------

Confusion Matrix

[[26 0 0 0 0 0 0 0]

[ 0 22 0 0 0 0 0 0]

[ 0 0 4 0 0 0 0 0]

[ 0 0 0 4 0 0 0 0]

[ 0 0 0 0 5 0 0 0]

[ 0 0 0 0 0 7 0 0]

[ 0 0 0 0 0 0 11 0]

[ 0 0 0 0 0 0 0 2]]

----------------------------------------------------------

Classification Report

precision recall f1-score support

0 1.00 1.00 1.00 26

1 1.00 1.00 1.00 22

2 1.00 1.00 1.00 4

3 1.00 1.00 1.00 4

4 1.00 1.00 1.00 5

5 1.00 1.00 1.00 7

6 1.00 1.00 1.00 11

7 1.00 1.00 1.00 2

accuracy 1.00 81

macro avg 1.00 1.00 1.00 81

weighted avg 1.00 1.00 1.00 81

* The Accuracy what we have got is 1.0

and the Mean of Cross Validation Score 0.9889, which is good in condition.

model**=**XGBClassifier(verbosity**=**0)

model**.**fit(xtrain,ytrain)

p**=**model**.**predict(xtest)

s**=**cross\_val\_score(model,x,y,cv**=**10)

**Hyperparameter Tuning**

In [71]:

**from** sklearn.model\_selection **import** RandomizedSearchCV

Random Forest

In [72]:

params**=**{'n\_estimators':[100, 300, 500, 700],

'min\_samples\_split':[1,2,3,4],

'min\_samples\_leaf':[1,2,3,4],

'max\_depth':[**None**,1,2,3,4,5,6,7,8,9,10,15,20,25,30,35,40]}

In [76]:

g**=**RandomizedSearchCV(RandomForestClassifier(),params,cv**=**10)

g**.**fit(xtrain,ytrain)

Out[79]:

RandomizedSearchCV(cv=10, estimator=RandomForestClassifier(),

param\_distributions={'max\_depth': [None, 1, 2, 3, 4, 5, 6, 7,

8, 9, 10, 15, 20, 25, 30,

35, 40],

'min\_samples\_leaf': [1, 2, 3, 4],

'min\_samples\_split': [1, 2, 3, 4],

'n\_estimators': [100, 300, 500, 700]})

In [78]:

print(g**.**best\_estimator\_)

print(g**.**best\_params\_)

print(g**.**best\_score\_)

RandomForestClassifier(max\_depth=30, min\_samples\_leaf=4, n\_estimators=500)

{'n\_estimators': 500, 'min\_samples\_split': 2, 'min\_samples\_leaf': 4, 'max\_depth': 30}

0.9947368421052631

m**=**RandomForestClassifier(max\_depth**=**20, min\_samples\_leaf**=**4, min\_samples\_split**=**4,n\_estimators**=**700)

m**.**fit(xtrain,ytrain)

p**=**m**.**predict(xtest)

score**=**cross\_val\_score(m,x,y,cv**=**10)

In [81]:

print('Accuracy',np**.**round(accuracy\_score(p,ytest),4))

print('----------------------------------------------------------')

print('Mean of Cross Validation Score',np**.**round(s**.**mean(),4))

print('----------------------------------------------------------')

print('Confusion Matrix')

print(confusion\_matrix(p,ytest))

print('----------------------------------------------------------')

print('Classification Report')

print(classification\_report(p,ytest))

Accuracy 1.0

----------------------------------------------------------

Mean of Cross Validation Score 0.9889

----------------------------------------------------------

Confusion Matrix

[[26 0 0 0 0 0 0 0]

[ 0 22 0 0 0 0 0 0]

[ 0 0 4 0 0 0 0 0]

[ 0 0 0 4 0 0 0 0]

[ 0 0 0 0 5 0 0 0]

[ 0 0 0 0 0 7 0 0]

[ 0 0 0 0 0 0 11 0]

[ 0 0 0 0 0 0 0 2]]

----------------------------------------------------------

Classification Report

precision recall f1-score support

0 1.00 1.00 1.00 26

1 1.00 1.00 1.00 22

2 1.00 1.00 1.00 4

3 1.00 1.00 1.00 4

4 1.00 1.00 1.00 5

5 1.00 1.00 1.00 7

6 1.00 1.00 1.00 11

7 1.00 1.00 1.00 2

accuracy 1.00 81

macro avg 1.00 1.00 1.00 81

weighted avg 1.00 1.00 1.00 81

* Herwe the Accuracy is 1.0 and the Mean of Cross Validation Score 0.9889, which is same for both the analysis

**Xgboost**

In [82]:

params**=**{'n\_estimators':[100,200,300,400,500],

'learning\_rate':[0.001,0.01,0.10,],

'subsample':[0.5,1],

'max\_depth':[1,2,3,4,5,6,7,8,9,10]}

In [ ]:

g**=**RandomizedSearchCV(XGBClassifier(),params,cv**=**10)

g**.**fit(xtrain,ytrain)

In [ ]:

print(g**.**best\_estimator\_)

print(g**.**best\_params\_)

print(g**.**best\_score\_)

**Finalizing the best Model**

In [ ]:

model**=**XGBClassifier(max\_depth**=**2,learning\_rate**=**0.01,n\_estimators**=**500,subsample**=**1)

model**.**fit(xtrain,ytrain)

p**=**model**.**predict(xtest)

score**=**cross\_val\_score(model,x,y,cv**=**10)

**Saving the Model**

In [85]:

**import** joblib

joblib**.**dump(model,'Retention.obj')

Out[85]:

['Retention.obj']

**Conclusion**

**\***All the websites were **not** equally preferred by online customers**.**

**\***Amazon was the most preferred followed by Flipkart**.**

**\***These two companies are most trusted **in** the industry **and** hence, have

a huge reliability**.** Also, the sellers listed on these websites are generally **from** Tier 1 cities **as** compared to Snapdeal **and** PayTM which have more sellers **from** tier 2 **and** 3 cities**.**

**\***The cost of the product, the reliability of the E**-**commerce company **and** the **return** policiesall play an equally important role **in** deciding the buying behaviour of online customers**.**

**\***The cost **is** an important factor **as** it was the basic criteria used by online retailers to attract customers**.** The reliability of the E**-**commerce company **is** also important, **as** it **is** even required **in** offline retail**.** It **is** important because customers are paying online, so they need to be sure of security of the online transaction**.** The **return** policies are important

because **in** online retail customer does **not** get to feel the product**.** Thus, he wants to be surethat it will be possible to **return** the product **if** he does **not** like it **in** real**.** Whereas, the logistics factor, which included Cash on delivery option, One day delivery **and** the quality of packaging plays a secondary role **in** this process though these are Must**-**be**-**quality**.**

**\***This **is** so because these all does **not** interfere **with** the real product **and** people believe that this **is** the basic value that E**-**commerce websitesprovide**.**

**\***Also, these websites have the most lenient **return** policies **as** compared to others **and** also the time required to process a **return** **is** low**.**

**-------------------------------------**End**-----------------------------**