import pandas as pd

import numpy as np import matplotlib.pyplot as plt %matplotlib inline

In [88]:
```
import io
%cd "C:\Users\uttej\OneDrive\Desktop\python 2"
```

```
C:\Users\uttej\OneDrive\Desktop\python 2
```

In [89]:
```
dresses=pd.read_csv("Attribute dataset.csv")
```

In [90]:
```
dresses.head(10)
```

Out[90]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	M
0	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleevless	empire	
1	1212192089	Casual	Low	0.0	L	Summer	o-neck	Petal	natural	mic
2	1190380701	vintage	High	0.0	L	Automn	o-neck	full	natural	p
3	966005983	Brief	Average	4.6	L	Spring	o-neck	full	natural	
4	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffo
5	1068332458	bohemian	Low	0.0	M	Summer	v-neck	sleevless	empire	
6	1220707172	Casual	Average	0.0	XL	Summer	o-neck	full	NaN	
7	1219677488	Novelty	Average	0.0	free	Automn	o-neck	short	natural	p
8	1113094204	Flare	Average	0.0	free	Spring	v-neck	short	empire	
9	985292672	bohemian	Low	0.0	free	Summer	v-neck	sleevless	natural	

In [91]:
```
dresses.tail(10)
```

Out[91]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Mat
490	641665398	Casual	Low	4.8	free	winter	bowneck	full	natural	pol
491	964917582	Casual	Average	5.0	L	Summer	o-neck	sleevless	natural	c
492	859922576	cute	Average	4.6	M	Spring	o-neck	halfsleeve	natural	a
493	817353671	bohemian	Low	4.6	free	Summer	o-neck	sleevless	natural	c
494	990559192	Brief	Average	4.7	M	winter	o-neck	halfsleeve	natural	a
495	713391965	Casual	Low	4.7	M	Spring	o-neck	full	natural	pol
496	722565148	Sexy	Low	4.3	free	Summer	o-neck	full	empire	c
497	532874347	Casual	Average	4.7	M	Summer	v-neck	full	empire	c
498	655464934	Casual	Average	4.6	L	winter	boat-neck	sleevless	empire	

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	waiseline	Mat
499	919930954	Casual	Low	4.4	free	Summer	v-neck	short	empire	c

In [92]:
```
dresses.shape
```

Out[92]:
```
(500, 14)
```

In [93]:
```
#missing value imputation
dresses.isnull().sum().sort_values(ascending=False)
```

Out[93]:
```
FabricType        266
Decoration        236
Material          128
Pattern Type      109
waiseline          87
NeckLine            3
Price               2
Season              2
SleeveLength        2
Dress_ID            0
Style               0
Rating              0
Size                0
Recommendation      0
dtype: int64
```

In [94]:
```
fc=['FabricType','Decoration','Material','Pattern Type','waiseline','NeckLine','Pric
for col in fc:
    freq=dresses[col].value_counts()
    print(freq)
```

```
chiffon       135
broadcloth     31
worsted        19
jersey         12
shiffon         9
sattin          6
batik           2
wollen          2
Corduroy        2
poplin          2
dobby           2
tulle           2
organza         1
knitted         1
lace            1
knitting        1
flannel         1
woolen          1
flannael        1
satin           1
other           1
terry           1
Name: FabricType, dtype: int64
lace          70
sashes        42
beading       22
```

```
hollowout     21
applique      21
ruffles       17
bow           15
sequined      14
button         6
pockets        5
embroidary     5
flowers        4
ruched         3
crystal        3
rivet          3
feathers       2
draped         2
plain          2
none           2
pearls         1
pleat          1
Tiered         1
tassel         1
cascading      1
Name: Decoration, dtype: int64
cotton          152
polyster         99
silk             26
chiffonfabric    25
mix              12
nylon            10
rayon            10
milksilk          5
spandex           5
cashmere          4
acrylic           3
microfiber        3
lycra             3
linen             3
other             2
viscos            2
shiffon           2
lace              1
modal             1
knitting          1
sill              1
wool              1
model             1
Name: Material, dtype: int64
solid        203
print         71
patchwork     48
animal        21
striped       17
dot           14
geometric      5
plaid          3
leopard        3
floral         2
character      1
splice         1
leapord        1
none           1
Name: Pattern Type, dtype: int64
natural     304
empire      104
dropped       4
```

```
princess      1
Name: waiseline, dtype: int64
o-neck             271
v-neck             124
slash-neck          25
boat-neck           19
Sweetheart          14
turndowncollor      13
bowneck             10
peterpan-collor      6
sqare-collor         5
open                 3
Scoop                2
ruffled              1
mandarin-collor      1
sweetheart           1
halter               1
backless             1
Name: NeckLine, dtype: int64
Average      252
Low          129
low           45
Medium        30
very-high     21
high          15
High           6
Name: Price, dtype: int64
Summer    159
Spring    122
Winter     99
Automn     61
winter     46
Autumn      8
spring      2
summer      1
Name: Season, dtype: int64
sleevless         223
full               97
short              96
halfsleeve         35
threequarter       17
thressqatar        10
sleeveless          5
sleeevless          3
capsleeves          3
cap-sleeves         2
urndowncollor       1
half                1
turndowncollor      1
threequater         1
Petal               1
butterfly           1
sleveless           1
Name: SleeveLength, dtype: int64
Casual      232
Sexy         69
party        51
cute         45
vintage      25
bohemian     24
Brief        18
work         17
Novelty       8
sexy          7
```

```
Flare         2
OL            1
fashion       1
Name: Style, dtype: int64
M        177
free     173
L         96
S         37
XL        15
small      1
s          1
Name: Size, dtype: int64
```

In [95]:
```
dresses.describe()
```

Out[95]:

	Dress_ID	Rating	Recommendation
count	5.000000e+02	500.000000	500.000000
mean	9.055417e+08	3.528600	0.420000
std	1.736190e+08	2.005364	0.494053
min	4.442820e+08	0.000000	0.000000
25%	7.673164e+08	3.700000	0.000000
50%	9.083296e+08	4.600000	0.000000
75%	1.039534e+09	4.800000	1.000000
max	1.253973e+09	5.000000	1.000000

In [96]:
```
dresses.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Dress_ID        500 non-null    int64  
 1   Style           500 non-null    object 
 2   Price           498 non-null    object 
 3   Rating          500 non-null    float64
 4   Size            500 non-null    object 
 5   Season          498 non-null    object 
 6   NeckLine        497 non-null    object 
 7   SleeveLength    498 non-null    object 
 8   waiseline       413 non-null    object 
 9   Material        372 non-null    object 
 10  FabricType      234 non-null    object 
 11  Decoration      264 non-null    object 
 12  Pattern Type    391 non-null    object 
 13  Recommendation  500 non-null    int64  
dtypes: float64(1), int64(2), object(11)
memory usage: 54.8+ KB
```

In [97]:
```
dresses.duplicated().sum()
```

Out[97]:
```
0
```

In [98]:

```
dresses.value_counts()
```

Out[98]:

```
Dress_ID    Style   Price    Rating  Size  Season  NeckLine  SleeveLength  waiseline
Material       FabricType  Decoration  Pattern Type  Recommendation
511503677   Casual  Low      4.4     M     Summer  o-neck    halfsleeve    empire
chiffonfabric  chiffon     bow         solid         0                 1
1033144189  cute    Average  4.7     M     Spring  o-neck    full          natural
rayon          chiffon     beading     solid         1                 1
1027818824  Casual  Medium   0.0     M     Spring  v-neck    threequarter  natural
silk           chiffon     ruffles     solid         1                 1
1026634314  Casual  Average  4.4     L     Automn  o-neck    sleevless     natural
linen          chiffon     ruffles     animal        0                 1
1024954013  Casual  Average  4.7     L     Summer  o-neck    sleevless     natural
silk           chiffon     beading     solid         0                 1

..
857432541   Casual  Low      4.7     free  Spring  o-neck    halfsleeve    natural
silk           chiffon     sashes      print         0                 1
851945460   Casual  Average  4.6     L     Automn  o-neck    sleevless     empire
polyster       chiffon     ruffles     solid         0                 1
845853510   Casual  Average  4.5     L     Summer  o-neck    short         natural
cotton         jersey      pockets     solid         0                 1
832391864   Casual  Average  0.0     L     Winter  open      short         natural
mix            poplin      applique    solid         1                 1
1249825438  Sexy    Average  0.0     free  Autumn  o-neck    full          natural
polyster       sattin      hollowout   solid         0                 1
Length: 99, dtype: int64
```

In [99]:

```
dresses.value_counts(normalize=True)
```

Out[99]:

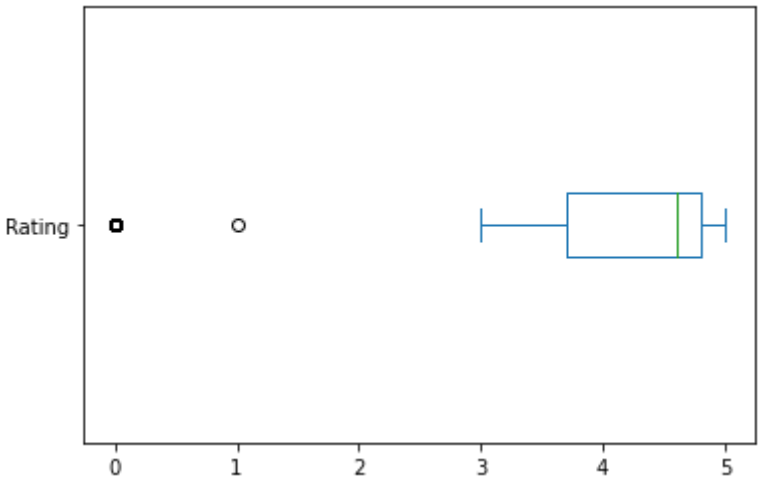
```
Dress_ID    Style   Price    Rating  Size  Season  NeckLine  SleeveLength  waiseline
Material       FabricType  Decoration  Pattern Type  Recommendation
511503677   Casual  Low      4.4     M     Summer  o-neck    halfsleeve    empire
chiffonfabric  chiffon     bow         solid         0                 0.010101
1033144189  cute    Average  4.7     M     Spring  o-neck    full          natural
rayon          chiffon     beading     solid         1                 0.010101
1027818824  Casual  Medium   0.0     M     Spring  v-neck    threequarter  natural
silk           chiffon     ruffles     solid         1                 0.010101
1026634314  Casual  Average  4.4     L     Automn  o-neck    sleevless     natural
linen          chiffon     ruffles     animal        0                 0.010101
1024954013  Casual  Average  4.7     L     Summer  o-neck    sleevless     natural
silk           chiffon     beading     solid         0                 0.010101

...
857432541   Casual  Low      4.7     free  Spring  o-neck    halfsleeve    natural
silk           chiffon     sashes      print         0                 0.010101
851945460   Casual  Average  4.6     L     Automn  o-neck    sleevless     empire
polyster       chiffon     ruffles     solid         0                 0.010101
845853510   Casual  Average  4.5     L     Summer  o-neck    short         natural
cotton         jersey      pockets     solid         0                 0.010101
832391864   Casual  Average  0.0     L     Winter  open      short         natural
mix            poplin      applique    solid         1                 0.010101
1249825438  Sexy    Average  0.0     free  Autumn  o-neck    full          natural
polyster       sattin      hollowout   solid         0                 0.010101
Length: 99, dtype: float64
```

In [100…

```
#Eploratory Data analysis on numerical Variable
dresses.Rating.plot(kind="box",vert=False)
```

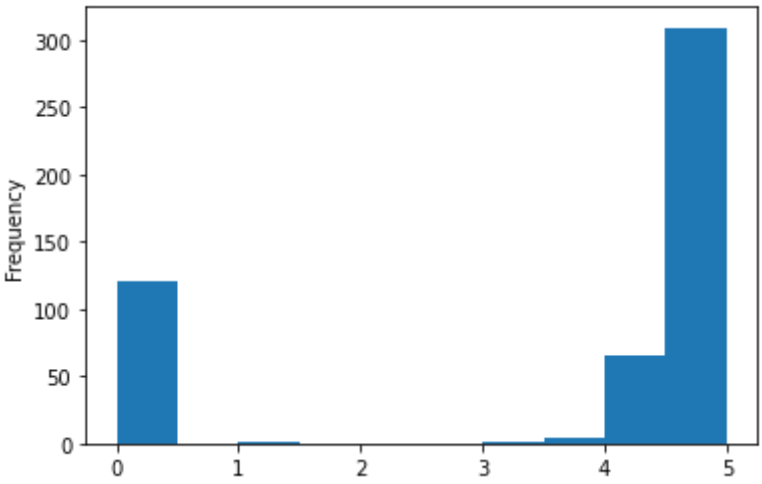
Out[100…

```
<AxesSubplot:>
```



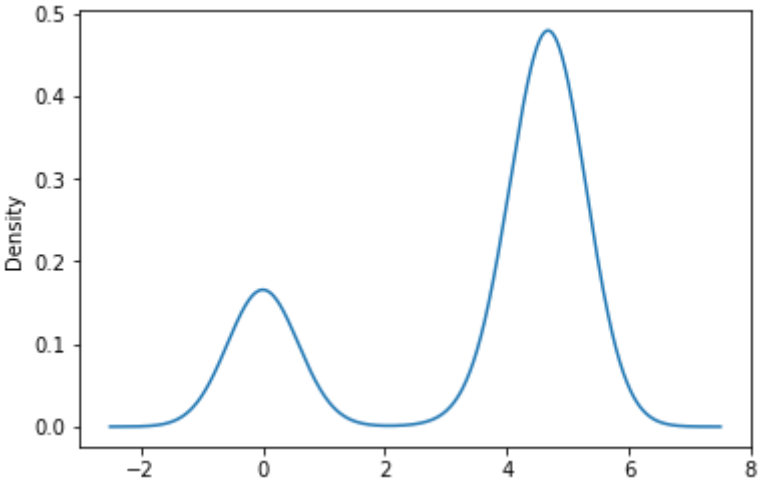
In [101…
```
dresses.Rating.plot(kind="hist")
```

Out[101…
```
<AxesSubplot:ylabel='Frequency'>
```



In [102…
```
dresses.Rating.plot(kind="density")
```

Out[102…
```
<AxesSubplot:ylabel='Density'>
```



In [103…
```
#replacing duplicates
dresses.FabricType=dresses.FabricType.replace('shiffon','chiffon')
dresses.FabricType=dresses.FabricType.replace('sattin','satin')
dresses.FabricType=dresses.FabricType.replace('wollen','woolen')
dresses.FabricType=dresses.FabricType.replace('knitting','knitted')
```

```
dresses.FabricType=dresses.FabricType.replace('flanneal','flannel')
dresses.Material=dresses.Material.replace('shiffon','chiffonfabric')
dresses.NeckLine=dresses.NeckLine.replace('sweetheart','Sweetheart')
dresses.Price=dresses.Price.replace('low','Low')
dresses.Price=dresses.Price.replace('high','High')
dresses.Price=dresses.Price.replace('Medium','Average')
dresses.Season=dresses.Season.replace('spring','Spring')
dresses.Season=dresses.Season.replace('summer','Summer')
dresses.Season=dresses.Season.replace('Automn','Autumn')
dresses.Season=dresses.Season.replace('winter','Winter')
dresses.SleeveLength=dresses.SleeveLength.replace('cap-sleeves','capsleeves')
dresses.SleeveLength=dresses.SleeveLength.replace('urndowncollor','turndowncollor')
dresses.SleeveLength=dresses.SleeveLength.replace('half','halfsleeve')
dresses['Style']=dresses['Style'].replace('sexy','Sexy')
dresses.Size=dresses.Size.replace(['small','s'],'S')
```

In [104…

```
dresses.describe()
```

Out[104…

	Dress_ID	Rating	Recommendation
count	5.000000e+02	500.000000	500.000000
mean	9.055417e+08	3.528600	0.420000
std	1.736190e+08	2.005364	0.494053
min	4.442820e+08	0.000000	0.000000
25%	7.673164e+08	3.700000	0.000000
50%	9.083296e+08	4.600000	0.000000
75%	1.039534e+09	4.800000	1.000000
max	1.253973e+09	5.000000	1.000000

In [105…

```
numericcols=dresses.select_dtypes(include=np.number)
objectcols=dresses.select_dtypes(include=['object'])
```

In [106…

```
numericcols.isnull().sum().sort_values(ascending=False)
```

Out[106…

```
Dress_ID          0
Rating            0
Recommendation    0
dtype: int64
```

In [107…

```
objectcols.isnull().sum().sort_values(ascending=False)
```

Out[107…

```
FabricType      266
Decoration      236
Material        128
Pattern Type    109
waiseline        87
NeckLine          3
Price             2
Season            2
SleeveLength      2
Style             0
Size              0
dtype: int64
```

In [108…
```
from sklearn.impute import SimpleImputer
```

In [109…
```
impute=SimpleImputer(strategy='most_frequent')
```

In [110…
```
objectcols_imputed=impute.fit_transform(objectcols)
```

In [111…
```
objectcols_imputed=pd.DataFrame(objectcols_imputed,columns=objectcols.columns)
```

In [112…
```
objectcols_imputed.isnull().sum().sort_values(ascending=False)
```

Out[112…
```
Style           0
Price           0
Size            0
Season          0
NeckLine        0
SleeveLength    0
waiseline       0
Material        0
FabricType      0
Decoration      0
Pattern Type    0
dtype: int64
```

In [113…
```
numericcols=numericcols.drop('Dress_ID',axis=1)
```

In [114…
```
dresses_imputed=pd.concat([objectcols_imputed,numericcols],axis=1)
```

In [115…
```
dresses_imputed.head()
```

Out[115…

	Style	Price	Size	Season	NeckLine	SleeveLength	waiseline	Material	FabricType	Dec
0	Sexy	Low	M	Summer	o-neck	sleevless	empire	cotton	chiffon	
1	Casual	Low	L	Summer	o-neck	Petal	natural	microfiber	chiffon	
2	vintage	High	L	Autumn	o-neck	full	natural	polyster	chiffon	
3	Brief	Average	L	Spring	o-neck	full	natural	silk	chiffon	em
4	cute	Low	M	Summer	o-neck	butterfly	natural	chiffonfabric	chiffon	

In [116…
```
#frequency counts on categorial variables
fc=['FabricType','Decoration','Material','Pattern Type','waiseline','NeckLine','Pric
for col in fc:
    freq=dresses[col].value_counts()
    print(freq)
```

```
chiffon       144
broadcloth     31
worsted        19
jersey         12
satin           7
```

```
woolen          3
dobby           2
Corduroy        2
tulle           2
knitted         2
poplin          2
batik           2
flannael        1
flannel         1
organza         1
lace            1
other           1
terry           1
Name: FabricType, dtype: int64
lace          70
sashes        42
beading       22
hollowout     21
applique      21
ruffles       17
bow           15
sequined      14
button         6
pockets        5
embroidary     5
flowers        4
ruched         3
crystal        3
rivet          3
feathers       2
draped         2
plain          2
none           2
pearls         1
pleat          1
Tiered         1
tassel         1
cascading      1
Name: Decoration, dtype: int64
cotton          152
polyster         99
chiffonfabric    27
silk             26
mix              12
nylon            10
rayon            10
milksilk          5
spandex           5
cashmere          4
acrylic           3
microfiber        3
lycra             3
linen             3
other             2
viscos            2
lace              1
modal             1
knitting          1
sill              1
wool              1
model             1
Name: Material, dtype: int64
solid        203
print         71
```

```
patchwork     48
animal        21
striped       17
dot           14
geometric      5
plaid          3
leopard        3
floral         2
character      1
splice         1
leapord        1
none           1
Name: Pattern Type, dtype: int64
natural     304
empire      104
dropped       4
princess      1
Name: waiseline, dtype: int64
o-neck             271
v-neck             124
slash-neck          25
boat-neck           19
Sweetheart          15
turndowncollor      13
bowneck             10
peterpan-collor      6
sqare-collor         5
open                 3
Scoop                2
ruffled              1
mandarin-collor      1
halter               1
backless             1
Name: NeckLine, dtype: int64
Average      282
Low          174
High          21
very-high     21
Name: Price, dtype: int64
Summer    160
Winter    145
Spring    124
Autumn     69
Name: Season, dtype: int64
sleevless         223
full               97
short              96
halfsleeve         36
threequarter       17
thressqatar        10
capsleeves          5
sleeveless          5
sleeevless          3
turndowncollor      2
Petal               1
butterfly           1
threequater         1
sleveless           1
Name: SleeveLength, dtype: int64
Casual      232
Sexy         76
party        51
cute         45
vintage      25
```

```
bohemian     24
Brief        18
work         17
Novelty       8
Flare         2
OL            1
fashion       1
Name: Style, dtype: int64
M       177
free    173
L        96
S        39
XL       15
Name: Size, dtype: int64
```

In [117…

```
#season wise Fabric type sold
pd.crosstab(dresses_imputed.Season,dresses_imputed.FabricType)
```

Out[117…

FabricType	Corduroy	batik	broadcloth	chiffon	dobby	flannael	flannel	jersey	knitted	lace	o
Season											
Autumn	0	0	3	61	1	0	0	0	0	0	
Spring	0	0	5	102	1	0	0	3	0	0	
Summer	2	2	11	129	0	0	0	6	0	0	
Winter	0	0	12	118	0	1	1	3	2	1	

In [118…

```
#price category of different variables
pd.crosstab(dresses_imputed.Material,dresses_imputed.Price)
```

Out[118…

Price	Average	High	Low	very-high
Material				
acrylic	2	0	1	0
cashmere	4	0	0	0
chiffonfabric	10	0	16	1
cotton	160	11	99	10
knitting	0	0	1	0
lace	0	0	1	0
linen	3	0	0	0
lycra	1	0	1	1
microfiber	1	0	1	1
milksilk	2	0	3	0
mix	8	1	3	0
modal	1	0	0	0
model	0	0	1	0
nylon	4	2	4	0

Price	Average	High	Low	very-high
Material				
other	1	0	1	0
polyster	69	4	21	5
rayon	6	1	2	1
silk	9	2	14	1
sill	0	0	1	0
spandex	3	0	2	0
viscos	0	0	1	1
wool	0	0	1	0

In [119…
```
#price category of different styles
pd.crosstab(dresses_imputed.Style,dresses_imputed.Price)
```

Out[119…

Price	Average	High	Low	very-high
Style				
Brief	13	0	5	0
Casual	130	6	94	2
Flare	2	0	0	0
Novelty	8	0	0	0
OL	1	0	0	0
Sexy	34	1	40	1
bohemian	11	0	12	1
cute	25	3	17	0
fashion	1	0	0	0
party	25	9	0	17
vintage	19	2	4	0
work	15	0	2	0

In [120…
```
#average rating of different styles
dresses_imputed.Rating.groupby(dresses_imputed.Style).mean()
```

Out[120…
```
Style
Brief       4.111111
Casual      3.576724
Flare       0.000000
Novelty     2.375000
OL          0.000000
Sexy        3.418421
bohemian    3.912500
cute        3.306667
fashion     4.000000
party       3.711765
vintage     3.168000
```

```
work        3.911765
Name: Rating, dtype: float64
```

In [121…
```
#TTest-there are no 2 categorial variables with only two levels
```

In [126…
```
#anova single factor
#null-there is no significant difference in average rating in price
#alt-there is no significant difference in average rating of price

price_low=dresses_imputed[dresses_imputed.Price=='Low']
price_average=dresses_imputed[dresses_imputed.Price=='Average']
price_high=dresses_imputed[dresses_imputed.Price=='High']
price_veryhigh=dresses_imputed[dresses_imputed.Price=='Very-High']


from scipy.stats import f_oneway
f_oneway(price_low.Rating,price_average.Rating,price_high.Rating,price_veryhigh.Rati
```

Out[126…
```
F_onewayResult(statistic=nan, pvalue=nan)
```

In [128…
```
#chi square test
#null- there is no association
#alt - there is a association

#test bull there is no association between style and price
from scipy.stats import chi2_contingency

chi2_contingency(pd.crosstab(dresses_imputed.Style,dresses_imputed.Price))
```

Out[128…
```
(195.48402456280908,
 5.705540832857158e-25,
 33,
 array([[1.02240e+01, 7.56000e-01, 6.26400e+00, 7.56000e-01],
        [1.31776e+02, 9.74400e+00, 8.07360e+01, 9.74400e+00],
        [1.13600e+00, 8.40000e-02, 6.96000e-01, 8.40000e-02],
        [4.54400e+00, 3.36000e-01, 2.78400e+00, 3.36000e-01],
        [5.68000e-01, 4.20000e-02, 3.48000e-01, 4.20000e-02],
        [4.31680e+01, 3.19200e+00, 2.64480e+01, 3.19200e+00],
        [1.36320e+01, 1.00800e+00, 8.35200e+00, 1.00800e+00],
        [2.55600e+01, 1.89000e+00, 1.56600e+01, 1.89000e+00],
        [5.68000e-01, 4.20000e-02, 3.48000e-01, 4.20000e-02],
        [2.89680e+01, 2.14200e+00, 1.77480e+01, 2.14200e+00],
        [1.42000e+01, 1.05000e+00, 8.70000e+00, 1.05000e+00],
        [9.65600e+00, 7.14000e-01, 5.91600e+00, 7.14000e-01]]))
```

In [130…
```
from sklearn.preprocessing import LabelEncoder
```

In [131…
```
le=LabelEncoder()
```

In [132…
```
dresses_imputed[['FabricType','Decoration','Material','Pattern Type','waiseline','Ne
```

In [133…
```
dresses_imputed.head()
```

Out[133…

	Style	Price	Size	Season	NeckLine	SleeveLength	waiseline	Material	FabricType	Decoration
0	5	2	1	2	7	8	1	3	3	20
1	1	2	0	2	7	0	2	8	3	20
2	10	1	0	0	7	3	2	15	3	12
3	0	0	0	1	7	3	2	17	3	8
4	7	2	1	2	7	1	2	2	3	3

In [134…
```
#machine learning modules
y=dresses_imputed.Recommendation
x=dresses_imputed.drop('Recommendation',axis=1)
```

In [136…
```
print(y.shape)
print(x.shape)
```

```
(500,)
(500, 12)
```

In [137…
```
#binary logistic regression
from sklearn.linear_model import LogisticRegression
```

In [138…
```
lr=LogisticRegression()
```

In [139…
```
lrmodel=lr.fit(x,y)
```

In [140…
```
lrmodel.score(x,y)
```

Out[140…
```
0.612
```

In [141…
```
lrpredict=lrmodel.predict(x)
```

In [142…
```
from sklearn.metrics import classification_report
```

In [143…
```
print(classification_report(y,lrpredict))
```

```
              precision    recall  f1-score   support

           0       0.62      0.88      0.73       290
           1       0.60      0.24      0.34       210

    accuracy                           0.61       500
   macro avg       0.61      0.56      0.53       500
weighted avg       0.61      0.61      0.56       500
```

In [145…
```
#decesion tree classifier
```

```
from sklearn.tree import DecisionTreeClassifier
```

In [146…
```
tree=DecisionTreeClassifier(max_depth=8)
```

In [147…
```
treemodel=tree.fit(x,y)
```

In [148…
```
treemodel.score(x,y)
```

Out[148…
```
0.83
```

In [149…
```
treepredict=tree.predict(x)
```

In [150…
```
print(classification_report(y,treepredict))
```

```
              precision    recall  f1-score   support

           0       0.85      0.86      0.85       290
           1       0.80      0.79      0.80       210

    accuracy                           0.83       500
   macro avg       0.83      0.82      0.82       500
weighted avg       0.83      0.83      0.83       500
```

In [151…
```
#random forest classifier
from sklearn.ensemble import RandomForestClassifier
```

In [152…
```
rf=RandomForestClassifier(n_estimators=1000,max_depth=8)
```

In [153…
```
rfmodel=rf.fit(x,y)
```

In [155…
```
rfmodel.score(x,y)
```

Out[155…
```
0.902
```

In [156…
```
rfpredict=rfmodel.predict(x)
```

In [157…
```
print(classification_report(y,rfpredict))
```

```
              precision    recall  f1-score   support

           0       0.86      0.99      0.92       290
           1       0.98      0.79      0.87       210

    accuracy                           0.90       500
   macro avg       0.92      0.89      0.90       500
weighted avg       0.91      0.90      0.90       500
```

In [160…
```
#gradient boost classifier
from sklearn.ensemble import GradientBoostingClassifier
```

In [161…
```
gbc=GradientBoostingClassifier(n_estimators=1000)
```

In [162…
```
gbcmodel=gbc.fit(x,y)
```

In [163…
```
gbcmodel.score(x,y)
```

Out[163…
```
0.994
```

In [165…
```
gbcpredict=gbcmodel.predict(x)
```

In [166…
```
print(classification_report(y,gbcpredict))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       290
           1       1.00      0.99      0.99       210

    accuracy                           0.99       500
   macro avg       0.99      0.99      0.99       500
weighted avg       0.99      0.99      0.99       500
```

In [167…
```
#support vector machine
from sklearn.svm import SVC
```

In [168…
```
svc=SVC()
```

In [169…
```
svcmodel=svc.fit(x,y)
```

In [170…
```
svcmodel.score(x,y)
```

Out[170…
```
0.62
```

In [171…
```
svcpredict=svcmodel.predict(x)
```

In [172…
```
print(classification_report(y,svcpredict))
```

```
              precision    recall  f1-score   support

           0       0.61      0.98      0.75       290
           1       0.79      0.13      0.22       210

    accuracy                           0.62       500
   macro avg       0.70      0.55      0.48       500
weighted avg       0.69      0.62      0.53       500
```

In [173…
```
# K nearest neighbour
from sklearn.naive_bayes import GaussianNB
```

In [174…
```
nb=GaussianNB()
```

In [175…
```
nbmodel=nb.fit(x,y)
```

In [176…
```
nbmodel.score(x,y)
```

Out[176…
```
0.586
```

In [177…
```
nbpredict=nbmodel.predict(x)
```

In [178…
```
print(classification_report(y,nbpredict))
```

```
              precision    recall  f1-score   support

           0       0.60      0.84      0.70       290
           1       0.52      0.23      0.32       210

    accuracy                           0.59       500
   macro avg       0.56      0.54      0.51       500
weighted avg       0.57      0.59      0.54       500
```

In [179…
```
#neural network-MLP classifier
from sklearn.neural_network import MLPClassifier
```

In [180…
```
nn=MLPClassifier(hidden_layer_sizes=(100,50,20))
```

In [181…
```
nnmodel=nn.fit(x,y)
```

```
C:\Users\uttej\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_percep
tron.py:614: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reac
hed and the optimization hasn't converged yet.
  warnings.warn(
```

In [182…
```
nnmodel.score(x,y)
```

Out[182…
```
0.928
```

In [183…
```
nnpredict=nnmodel.predict(x)
```

In [184…
```
print(classification_report(y,nnpredict))
```

```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       290
           1       0.95      0.87      0.91       210
```

```
              accuracy                           0.93       500
             macro avg       0.93      0.92      0.93       500
          weighted avg       0.93      0.93      0.93       500
```

In [ ]:

```
#gradient boosting clasasifier has got the highest value
```