Airline Recommendation System - Data 602 project - Group 4

Import the required Libraries

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Read the data

In [5]:

```
data = pd.read_csv("Airline_data.csv")
```

In [6]:

```
data.head(1)
```

Out[6]:

	Unnamed: 0	Unnamed: 0.1	airline_name	author	author_country	content	cabin_flown	overall
0	0	0	adria-airways	D Ito	Germany	Outbound flight FRA/PRN A319. 2 hours 10 min f...	Economy	

1 rows × 40 columns

In [7]:

```
data = data.drop(['Unnamed: 0.1','content','Class','Food and drink','Seat com
                  'Inflight entertainment','Departure Delay in Minutes','Arri
```

In [8]:

```
data.head(5)
```

Out[8]:

	Unnamed: 0	airline_name	author	author_country	cabin_flown	overall_rating	seat_comfo
0	0	adria-airways	D Ito	Germany	Economy	7.0	
1	1	adria-airways	Ron Kuhlmann	United States	Business Class	10.0	
2	2	adria-airways	E Albin	Switzerland	Economy	9.0	
3	3	adria-airways	Tercon Bojan	Singapore	Business Class	8.0	
4	4	adria-airways	L James	Poland	Economy	4.0	

5 rows × 31 columns

In [9]:

```
data.columns
```

Out[9]:

```
Index(['Unnamed: 0', 'airline_name', 'author', 'author_country', 'cabin_flo
wn',
       'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
       'food_beverages_rating', 'inflight_entertainment_rating',
       'value_money_rating', 'Month', 'Year', 'recommended', 'id', 'Gende
r',
       'Customer Type', 'Age', 'Type of Travel', 'Flight Distance',
       'Inflight wifi service', 'Departure/Arrival time convenient',
       'Ease of Online booking', 'Gate location', 'Online boarding',
       'On-board service', 'Leg room service', 'Baggage handling',
       'Checkin service', 'Inflight service', 'Cleanliness'],
      dtype='object')
```

In [10]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27284 entries, 0 to 27283
Data columns (total 31 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Unnamed: 0                         27284 non-null  int64  
 1   airline_name                       27284 non-null  object 
 2   author                             27284 non-null  object 
 3   author_country                     27284 non-null  object 
 4   cabin_flown                        27284 non-null  object 
 5   overall_rating                     27284 non-null  float64
 6   seat_comfort_rating                27284 non-null  float64
 7   cabin_staff_rating                 27284 non-null  float64
 8   food_beverages_rating              27284 non-null  float64
 9   inflight_entertainment_rating      27284 non-null  float64
 10  value_money_rating                 27284 non-null  float64
 11  Month                              27284 non-null  int64  
 12  Year                               27284 non-null  int64  
 13  recommended                        27284 non-null  int64  
 14  id                                 27284 non-null  int64  
 15  Gender                             27284 non-null  object 
 16  Customer Type                      27284 non-null  object 
 17  Age                                27284 non-null  int64  
 18  Type of Travel                     27284 non-null  object 
 19  Flight Distance                    27284 non-null  int64  
 20  Inflight wifi service              27284 non-null  int64  
 21  Departure/Arrival time convenient  27284 non-null  int64  
 22  Ease of Online booking             27284 non-null  int64  
 23  Gate location                      27284 non-null  int64  
 24  Online boarding                    27284 non-null  int64  
 25  On-board service                   27284 non-null  int64  
 26  Leg room service                   27284 non-null  int64  
 27  Baggage handling                   27284 non-null  int64  
 28  Checkin service                    27284 non-null  int64  
 29  Inflight service                   27284 non-null  int64  
 30  Cleanliness                        27284 non-null  int64  
dtypes: float64(6), int64(18), object(7)
memory usage: 6.5+ MB
```

In [11]:

```
data.describe()
```

Out[11]:

```
        Unnamed: 0  overall_rating  seat_comfort_rating  cabin_staff_rating  food_beverages_rat

count  27284.000000   27284.000000         27284.000000        27284.000000          27284.000
mean   13641.500000       6.067879             3.259566            3.522944              3.016
std     7876.356709       3.216066             1.351689            1.460053              1.515
min        0.000000       1.000000             0.000000            0.000000              0.000
25%     6820.750000       3.000000             2.000000            2.000000              2.000
50%    13641.500000       7.000000             4.000000            4.000000              3.000
75%    20462.250000       9.000000             4.000000            5.000000              4.000
max    27283.000000      10.000000             5.000000            5.000000              5.000

8 rows × 24 columns
```

In [12]:
```
data.isna().sum()
```

Out[12]:
```
Unnamed: 0                           0
airline_name                         0
author                               0
author_country                       0
cabin_flown                          0
overall_rating                       0
seat_comfort_rating                  0
cabin_staff_rating                   0
food_beverages_rating                0
inflight_entertainment_rating        0
value_money_rating                   0
Month                                0
Year                                 0
recommended                          0
id                                   0
Gender                               0
Customer Type                        0
Age                                  0
Type of Travel                       0
Flight Distance                      0
Inflight wifi service                0
Departure/Arrival time convenient    0
Ease of Online booking               0
Gate location                        0
Online boarding                      0
On-board service                     0
Leg room service                     0
Baggage handling                     0
Checkin service                      0
Inflight service                     0
Cleanliness                          0
dtype: int64
```

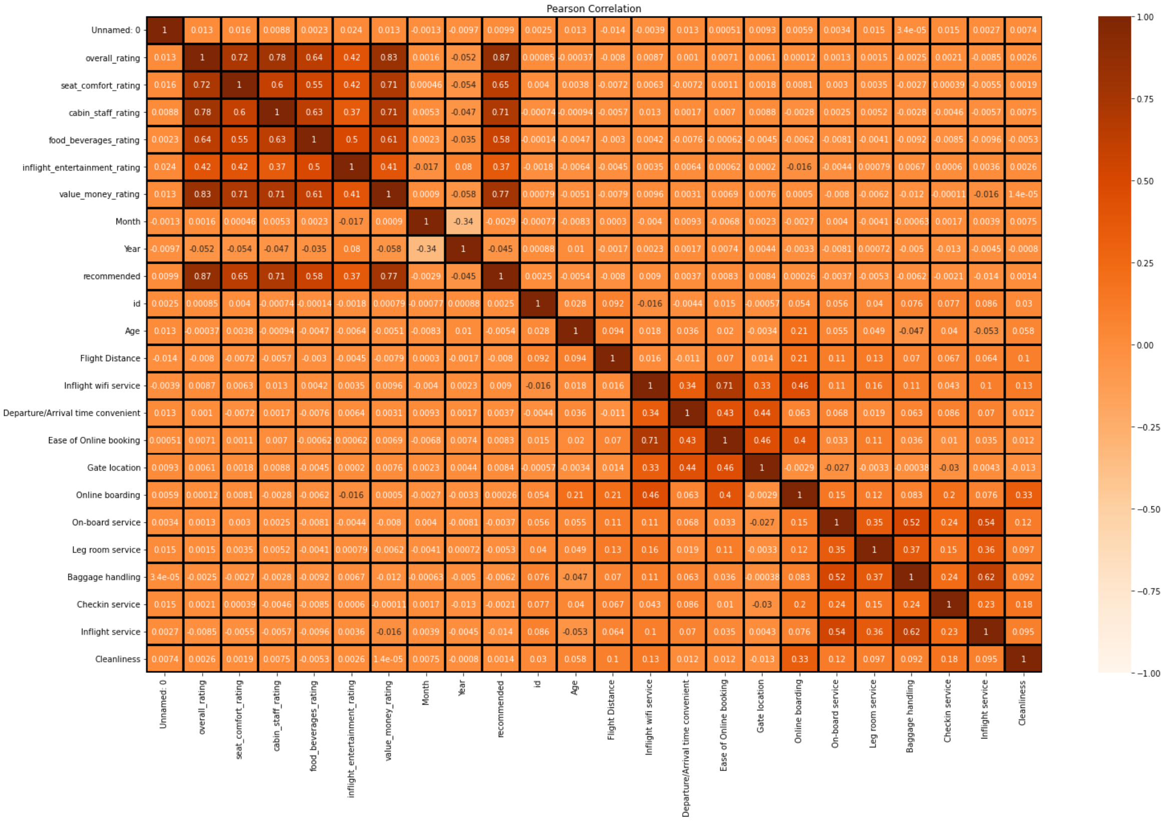
Data Exploration

Correlation Plot

In [13]:

```
import matplotlib.pyplot as plt
import seaborn as sns
corr_Pearson = data.corr(method='pearson')

figure = plt.figure(figsize=(25,15))
sns.heatmap(corr_Pearson,vmin=-1,vmax=+1,cmap='Oranges',annot=True,
            linewidths=2,linecolor = 'black')
plt.title('Pearson Correlation')
plt.show()
```



Pie chart

In [14]:

```
import plotly.express as px
air2 = data["recommended"].value_counts().reset_index()
fig = px.pie(air2,values="recommended",names="index",width=400, height=400)
fig.show()
```

In [15]:

```
gender = data["Gender"].value_counts().reset_index()
fig = px.pie(gender,values="Gender",names="index",width=400, height=400)
fig.show()
```

In [16]:

```
data.select_dtypes(include=['object']).columns
```

Out[16]:

```
Index(['airline_name', 'author', 'author_country', 'cabin_flown', 'Gender',
       'Customer Type', 'Type of Travel'],
      dtype='object')
```

In [17]:

```
air3 = data.groupby("cabin_flown")["author"].count().reset_index()
import plotly.graph_objects as go
fig = go.Figure(data=[go.Pie(labels=air3["cabin_flown"], values=air3["author"
fig.show()
```

In [18]:

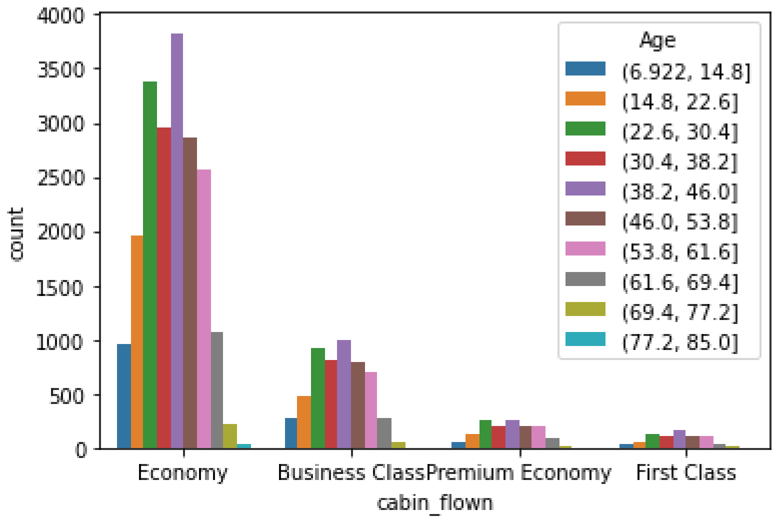
```
df = data.drop(['Unnamed: 0'],axis=1)
```

In [19]:

```
import seaborn as sns
sns.countplot(x='cabin_flown', hue=pd.cut(df['Age'],10),data=df)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fedc12fcb50>
```



In [20]:

```
air1 = data.groupby("airline_name")["author"].count().reset_index().sort_valu
a = list(air1["airline_name"])
df2 = data[data['airline_name'].isin(a)]
```

In [21]:

```
import plotly.express as px
fig = px.bar(air1, x='airline_name', y='author')
fig.show()
```

number of flights flown in each month

In [22]:

```
df.groupby('Month')['airline_name'].agg('count').sort_values()
```

Out[22]:

```
Month
11    1799
2     1890
8     1942
12    1952
9     2006
10    2259
3     2320
4     2331
6     2446
5     2592
7     2819
1     2928
Name: airline_name, dtype: int64
```

Which airline has received the most reiews by the customers

In [23]:

```
airline_counts = pd.DataFrame(df["airline_name"].value_counts())
airline_counts.sort_values("airline_name", ascending=False).head(10)
```

Out[23]:

	airline_name
british-airways	855
united-airlines	803
air-canada-rouge	703
emirates	685
lufthansa	586
american-airlines	579
qantas-airways	576
etihad-airways	512
qatar-airways	491
cathay-pacific-airways	491

Average rating for each airline

In [24]:

```
ratings = pd.DataFrame(df.groupby('airline_name')['overall_rating'].mean())
ratings.head()
```

Out[24]:

	overall_rating
airline_name	
adria-airways	7.705882
aegean-airlines	7.620690
aer-lingus	7.077703
aeroflot-russian-airlines	6.682051
aerogal-aerolineas-galapagos	8.500000

Count of rating given to each airline

In [25]:

```
ratings['num of ratings'] = pd.DataFrame(df.groupby('airline_name')['overall_
ratings.head()
```

Out[25]:

	overall_rating	num of ratings
airline_name		
adria-airways	7.705882	17
aegean-airlines	7.620690	174
aer-lingus	7.077703	296
aeroflot-russian-airlines	6.682051	195
aerogal-aerolineas-galapagos	8.500000	2

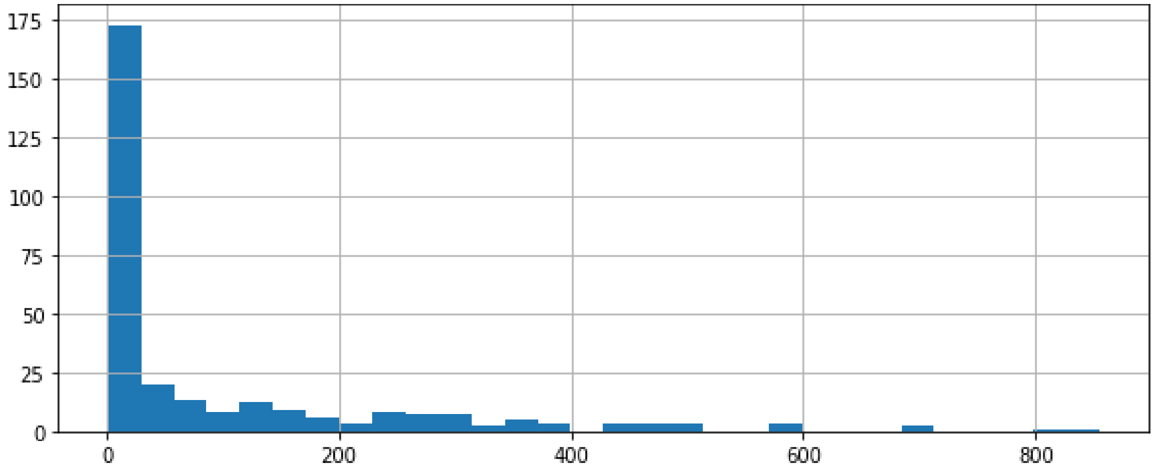
In [26]:
```
ratings.sort_values('num of ratings',ascending=False).head(10)
```

Out[26]:

	overall_rating	num of ratings
airline_name		
british-airways	5.905263	855
united-airlines	3.438356	803
air-canada-rouge	2.522048	703
emirates	6.265693	685
lufthansa	7.017065	586
american-airlines	3.696028	579
qantas-airways	7.008681	576
etihad-airways	4.910156	512
cathay-pacific-airways	6.916497	491
qatar-airways	7.313646	491

In [27]:
```
plt.figure(figsize=(10,4))
ratings['num of ratings'].hist(bins=30)
```

Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fedc0f0f190>
```

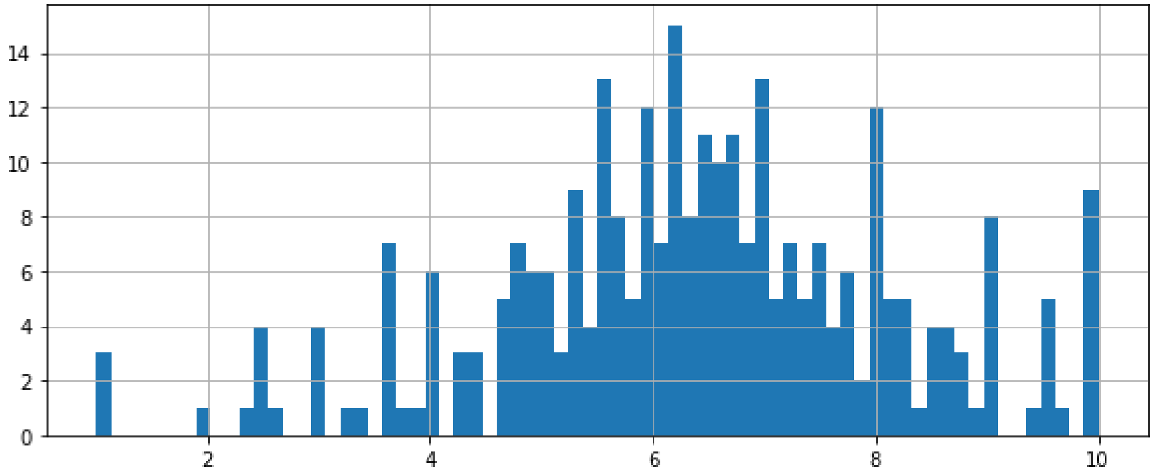


In [28]:

```
plt.figure(figsize=(10,4))
ratings['overall_rating'].hist(bins=70)
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fedc0e24590>
```



In [29]:

```
# list of numerical variables
numerical_features = [feature for feature in df.columns if df[feature].dtypes
print('Number of numerical variables: ', len(numerical_features))
df[numerical_features].head()# visualise the numerical variables
```

```
Number of numerical variables:  23
```

Out[29]:

	overall_rating	seat_comfort_rating	cabin_staff_rating	food_beverages_rating	inflight_enterta
0	7.0	4.0	4.0	4.0	
1	10.0	4.0	5.0	4.0	
2	9.0	5.0	5.0	4.0	
3	8.0	4.0	4.0	3.0	
4	4.0	4.0	2.0	1.0	

5 rows × 23 columns

In [30]:

```
categorical_features=[feature for feature in df.columns if df[feature].dtypes
categorical_features
```

Out[30]:

```
['airline_name',
 'author',
 'author_country',
 'cabin_flown',
 'Gender',
 'Customer Type',
 'Type of Travel']
```

In [31]:

```
review_spread1 = data.groupby("author_country")["overall_rating"].count().res
review_spread1.head(10)
```

Out[31]:

	author_country	overall_rating
133	United Kingdom	6275
134	United States	4967
5	Australia	3931
24	Canada	2625
47	Germany	885
114	Singapore	528
93	New Zealand	489
56	India	431
91	Netherlands	364
45	France	350

In [32]:

```
import plotly.express as px
fig = px.bar(review_spread1, x='author_country', y='overall_rating')
fig.show()
```

In [33]:

```
cabin_spread = data.groupby("cabin_flown")["overall_rating"].count().reset_ir
cabin_spread.head(10)
```

Out[33]:

	cabin_flown	overall_rating
1	Economy	19830
0	Business Class	5329
3	Premium Economy	1374
2	First Class	751

In [34]:

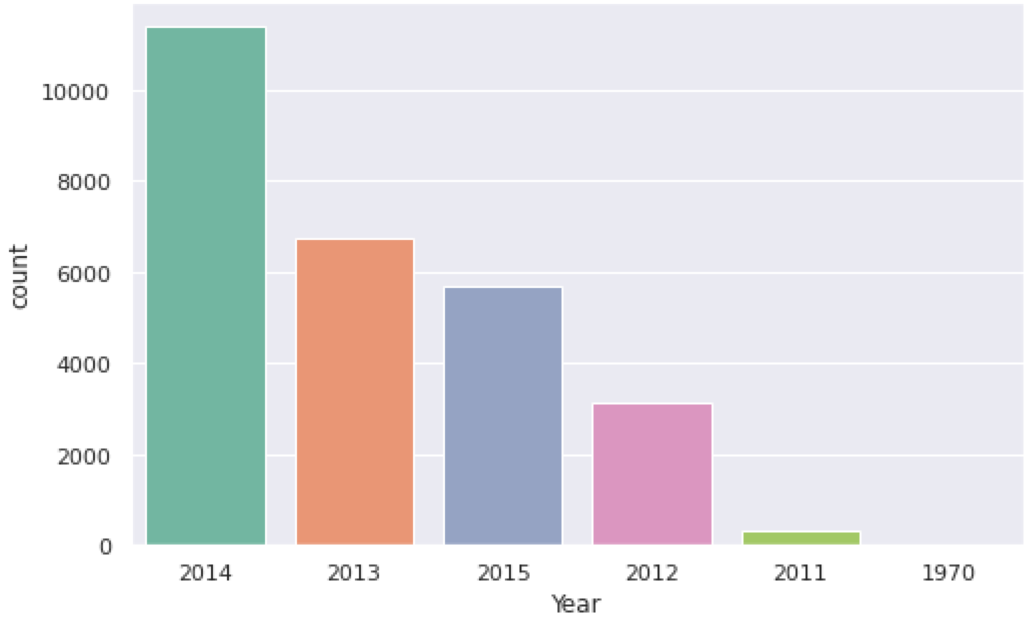
```
plt.figure(figsize=(8,5))
import plotly.express as px
fig = px.bar(cabin_spread, x='cabin_flown', y='overall_rating')
fig.show()
```

```
<Figure size 576x360 with 0 Axes>
```

Year wise analysis

In [35]:

```
plt.figure(figsize=(8,5))
sns.set(style="darkgrid")
ax = sns.countplot(x="Year", data=data, palette="Set2", order=data['Year']
                   .value_counts().index[0:15])
```



In [36]:

```
data['id_cus'] = data.groupby(['author']).ngroup()
```

In [37]:

```
data['airline_id'] = data[['airline_name']]
```

In [38]:

```
col = ['airline_id']
le= LabelEncoder()
data[col] = data[col].apply(le.fit_transform)
```

Cosine

In [39]:

```
airline_features_df=data.pivot_table(index='airline_name',columns='id_cus',va
airline_features_df.head()
```

Out[39]:

id_cus	0	1	2	3	4	5	6	7	8	9	...	19624	19625	19626	19627
airline_name															
adria-airways	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
aegean-airlines	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
aer-lingus	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
aeroflot-russian-airlines	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
aerogal-aerolineas-galapagos	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

5 rows × 19634 columns

In [40]:

```
from scipy.sparse import csr_matrix
airline_features_df_matrix = csr_matrix(airline_features_df.values)
from sklearn.neighbors import NearestNeighbors
model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
model_knn.fit(airline_features_df_matrix)
```

Out[40]:

```
NearestNeighbors(algorithm='brute', metric='cosine')
```

In [41]:

```
query_index = np.random.choice(airline_features_df.shape[0])
print(query_index)
distances, indices = model_knn.kneighbors(airline_features_df.iloc[query_inde
```

```
139
```

In [42]:

```
for i in range(0, len(distances.flatten())):
    if i == 0:
        print('Recommendations for {0}:\n'.format(airline_features_df.index[q
    else:
        print('{0}: {1}:'.format(i, airline_features_df.index[indices.flatter
```

```
Recommendations for hong-kong:

1: air-china:
2: dragonair:
3: eva-air:
4: cathay-pacific-airways:
5: malaysia-airlines:
```

In [43]:

```
col = df.columns.tolist()
```

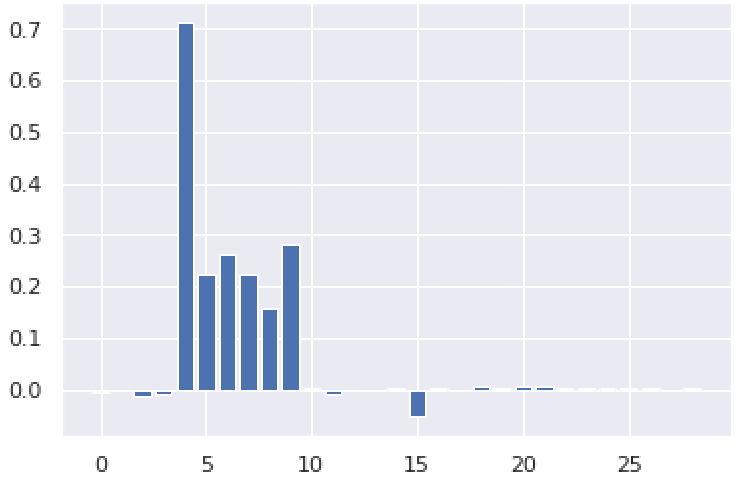
In [44]:

```
le= LabelEncoder()
df[col] = df[col].apply(le.fit_transform)
```

Apply the model and get the feature importance

In [45]:

```
# logistic regression for feature importance
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot
X = df.drop(['recommended'], axis=1)
y = df['recommended']
# define the model
model = LogisticRegression()
# fit the model
model.fit(X, y)
# get importance
importance = model.coef_[0]
# summarize feature importance
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```



In [46]:

```
from sklearn.datasets import make_classification
X = df.drop(['recommended'], axis=1)
y = df['recommended']
```

Split the dataset into training and testing dataset

In [47]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,y,test_size = 0.25, rar
```

Apply Gaussian algorithm to our data

In [48]:

```python
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
model=GaussianNB()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.9417973904119631
```

In [49]:

```python
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      2725
           1       0.96      0.95      0.95      4096

    accuracy                           0.94      6821
   macro avg       0.94      0.94      0.94      6821
weighted avg       0.94      0.94      0.94      6821
```

Apply KNN algorithm to our data

In [50]:

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix,f1_score,accuracy_score
model = KNeighborsClassifier(n_neighbors = 5, p =2, metric="euclidean")
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.5370180325465475
```

In [51]:

```python
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.39      0.29      0.33      2725
           1       0.60      0.70      0.65      4096

    accuracy                           0.54      6821
   macro avg       0.49      0.50      0.49      6821
weighted avg       0.52      0.54      0.52      6821
```

Apply Random Forest algorithm to our data

In [52]:

```
from sklearn.ensemble import RandomForestClassifier
Rd = RandomForestClassifier(n_estimators=20)
Rd.fit(x_train,y_train)
y_pred = Rd.predict(x_test)
print("Accuracy: ",accuracy_score(y_test,y_pred))
```

```
Accuracy:  0.9463421785661926
```

In [53]:

```
print(classification_report(y_test,y_pred))
```

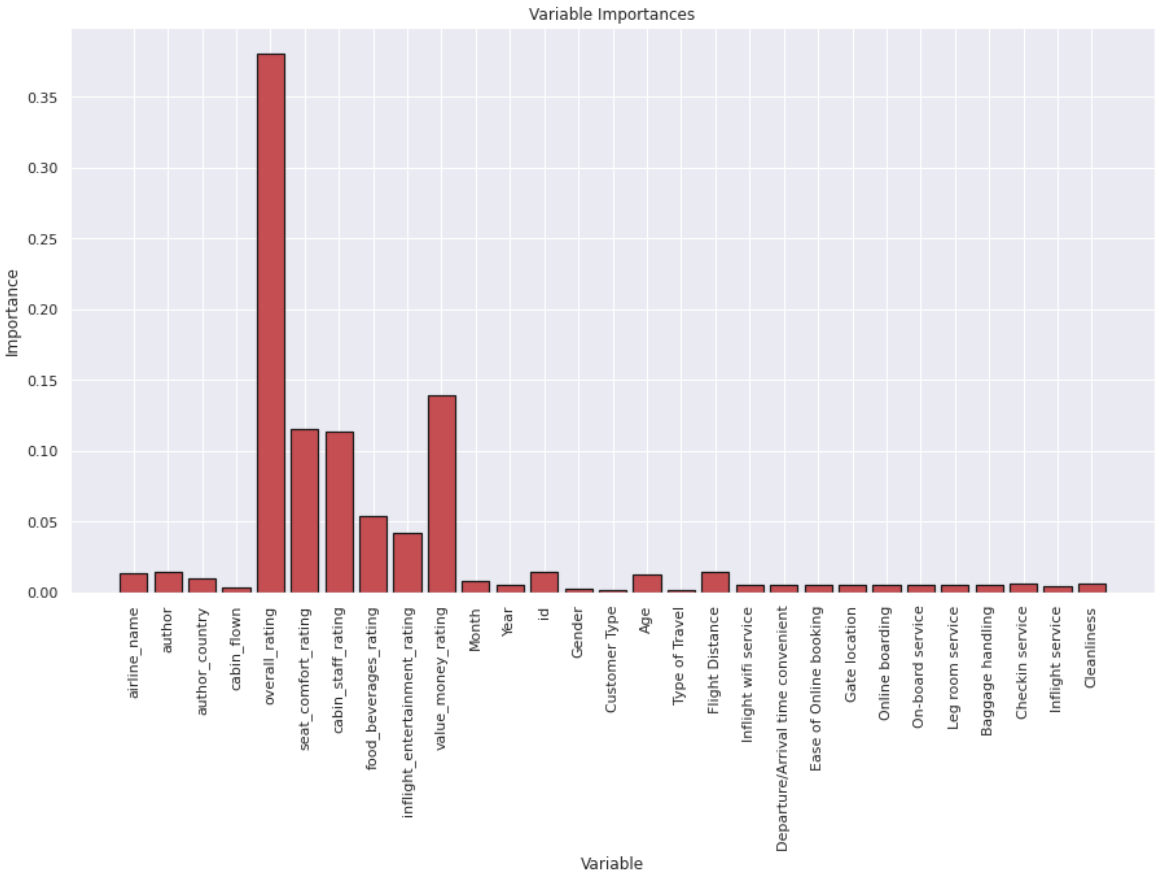
```
              precision    recall  f1-score   support

           0       0.93      0.94      0.93      2725
           1       0.96      0.95      0.96      4096

    accuracy                           0.95      6821
   macro avg       0.94      0.94      0.94      6821
weighted avg       0.95      0.95      0.95      6821
```

In [54]:
```
#Using Random Forest To get Feature Importance as It gives most accuracy
importances = list(Rd.feature_importances_)
x_values = list(range(len(importances)))
feature_list = X.columns
fig = plt.figure(figsize=(15,8))
plt.bar(x_values, importances, orientation = 'vertical', color = 'r', edgecol
plt.xticks(x_values, feature_list, rotation='vertical')
plt.ylabel('Importance'); plt.xlabel('Variable'); plt.title('Variable Importa
```

Out[54]:
```
Text(0.5, 1.0, 'Variable Importances')
```



Knowledge Based

In [55]:
```
new = data[['airline_name', 'author_country', 'cabin_flown',
       'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
       'food_beverages_rating', 'inflight_entertainment_rating',
       'value_money_rating','Age','recommended']]
```

In [56]:

```
new['Age'].describe()
```

Out[56]:

```
count    27284.000000
mean        39.361897
std         15.060640
min          7.000000
25%         27.000000
50%         40.000000
75%         51.000000
max         85.000000
Name: Age, dtype: float64
```

In [57]:

```
bins=[1,15,30,45,65,85]
labels = ['Childern','Youth', 'Adult', 'Middle Age', 'Senior']
new['Age'] = pd.cut(new['Age'], bins= bins, labels=labels)
```

In [58]:

```
new['count']=(new.groupby(['airline_name', 'cabin_flown','author_country','Ag
new
```

Out[58]:

	airline_name	author_country	cabin_flown	overall_rating	seat_comfort_rating	cabin_sta
0	adria-airways	Germany	Economy	7.0	4.0	
1	adria-airways	United States	Business Class	10.0	4.0	
2	adria-airways	Switzerland	Economy	9.0	5.0	
3	adria-airways	Singapore	Business Class	8.0	4.0	
4	adria-airways	Poland	Economy	4.0	4.0	
...	...	...	...	...	...	
27279	wizz-air	United Kingdom	Economy	7.0	3.0	
27280	wizz-air	United Kingdom	Economy	2.0	1.0	
27281	wizz-air	United Kingdom	Economy	1.0	2.0	
27282	wizz-air	United Kingdom	Economy	5.0	3.0	
27283	wizz-air	United Kingdom	Economy	1.0	2.0	

27284 rows × 12 columns

In [59]:

```
new = (new.groupby(['airline_name', 'cabin_flown','author_country','Age']).me
airline_age = new.sort_values(by='count', ascending=False)
airline_age
```

Out[59]:

				overall_rating	seat_comfort_rating	cabin_
airline_name	**cabin_flown**	**author_country**	**Age**			
air-canada-rouge	**Economy**	**Canada**	**Adult**	2.676829	1.506098	
			Middle Age	2.103226	1.270968	
			Youth	2.226562	1.367188	
sunwing-airlines	**Economy**	**Canada**	**Adult**	3.942308	2.086538	
united-airlines	**Economy**	**United States**	**Middle Age**	2.446602	2.077670	
...	**...**	**...**	**...**	...	...	
yangon-airways	**Premium Economy**	**Zimbabwe**	**Childern**	NaN	NaN	
			Youth	NaN	NaN	
			Adult	NaN	NaN	
			Middle Age	NaN	NaN	
			Senior	NaN	NaN	

829280 rows × 8 columns

In [60]:

```
new = new.dropna()
new
```

Out[60]:

				overall_rating	seat_comfort_rating	cabin_s
airline_name	**cabin_flown**	**author_country**	**Age**			
adria-airways	**Business Class**	**Singapore**	**Youth**	8.000000	4.000000	
			Adult	7.500000	3.500000	
		Turkey	**Youth**	7.000000	2.000000	
		United States	**Youth**	10.000000	4.000000	
	Economy	**Canada**	**Middle Age**	8.000000	4.000000	
...	**...**	**...**	**...**	...	...	
xl-airways-france	**Economy**	**United States**	**Adult**	8.500000	3.166667	
			Middle Age	6.333333	2.666667	
	Premium Economy	**Singapore**	**Middle Age**	3.000000	4.000000	
		United States	**Middle Age**	10.000000	3.000000	
yangon-airways	**Economy**	**Australia**	**Middle Age**	10.000000	5.000000	

8818 rows × 8 columns

In [61]:

```
#Airlines which rating count is more then 90%
m = new['count'].quantile(0.90)
print(m)
```

```
6.0
```

In [62]:

```
new = new.loc[new['count'] >= m]
new.shape
```

Out[62]:

```
(937, 8)
```

In [63]:

```
#Calculating weighting average of each airline important Ratings
for i in new[['overall_rating', 'seat_comfort_rating','cabin_staff_rating', '
    def weighted_rating(a, m=m, C=new[i].mean()):
        v = a['count']
        R = a[i]
        return (v/(v+m) * R) + (m/(m+v) * C)
    new[i +' score'] = new.apply(weighted_rating, axis =1)
```

In [64]:

```
new = new.reset_index()
```

In [65]:

```
#Airlines according to User Preferance
(new.loc[(new['cabin_flown'] == 'Economy') & (new['author_country'] == 'Unite
```

Out[65]:

	airline_name	cabin_flown	author_country	Age	overall_rating	seat_comfort_rating	cab
7	aegean-airlines	Economy	United States	Youth	10.000000	4.500000	
742	singapore-airlines	Economy	United States	Youth	8.428571	3.428571	
26	aeromexico	Economy	United States	Youth	7.900000	4.000000	
562	korean-air	Economy	United States	Youth	7.583333	3.583333	
550	klm-royal-dutch-airlines	Economy	United States	Youth	8.000000	3.666667	
334	china-southern-airlines	Economy	United States	Youth	7.750000	4.000000	
147	alaska-airlines	Economy	United States	Youth	7.368421	3.263158	
212	asiana-airlines	Economy	United States	Youth	7.461538	3.846154	
508	japan-airlines	Economy	United States	Youth	7.833333	3.666667	
199	ana-all-nippon-airways	Economy	United States	Youth	7.250000	2.916667	

Content Based

In [66]:

```
#Recommending Similar airline to Filtered airlines by Content Based Recommend
test = data[['airline_name','author_country','cabin_flown',
       'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
       'food_beverages_rating', 'inflight_entertainment_rating',
       'value_money_rating', 'Month', 'Year','Age', 'Flight Distance']]
```

In [67]:

```
test.head()
```

Out[67]:

	airline_name	author_country	cabin_flown	overall_rating	seat_comfort_rating	cabin_staff_ra
0	adria-airways	Germany	Economy	7.0	4.0	
1	adria-airways	United States	Business Class	10.0	4.0	
2	adria-airways	Switzerland	Economy	9.0	5.0	
3	adria-airways	Singapore	Business Class	8.0	4.0	
4	adria-airways	Poland	Economy	4.0	4.0	

In [68]:

```
test = (test.groupby(['airline_name', 'cabin_flown','author_country']).mean()
```

In [69]:

```
test = test.reset_index()
```

In [70]:

```
X = test.loc[:, ['author_country','cabin_flown',
       'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
       'food_beverages_rating', 'inflight_entertainment_rating',
       'value_money_rating', 'Month', 'Year','Age', 'Flight Distance']].value
```

In [71]:

```
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
ld= LabelEncoder()
X[:,0] = le.fit_transform(X[:, 0])
X[:,1] = ld.fit_transform(X[:, 1])
```

In [72]:

```
y = np.array(test.loc[(test['airline_name'] == 'air-india') & (test['cabin_fl
```

In [73]:

```
y = y[:,1:]
```

In [74]:

```
y[1]
```

Out[74]:

```
array(['Economy', 'Austria', '4.0', '1.0', '4.0', '4.0', '0.0', '2.0',
       '2.0', '2012.0', '22.0', '1276.0'], dtype='<U32')
```

In [75]:

```
y[:,0] = le.fit_transform(y[:,0])
y[:,1] = ld.fit_transform(y[:,1])
```

In [76]:

```
from sklearn.neighbors import NearestNeighbors
model = NearestNeighbors(n_neighbors = 4).fit(X)
result = model.kneighbors(y[[1]])
```

In [77]:

```
result = np.array(result[1])
result = result.flatten()
result
```

Out[77]:

```
array([ 503,  345, 3067, 3941])
```

In [78]:

```
#Recommendaded Airlines
test = test.loc[result]
test
```

Out[78]:

	airline_name	cabin_flown	author_country	overall_rating	seat_comfort_rating	cabin_sta
503	air-india	Economy	Austria	4.0	1.000000	4
345	air-china	Economy	Austria	4.0	2.000000	2
3067	malaysia-airlines	First Class	Australia	8.0	5.000000	5
3941	swiss-international-air-lines	Economy	Australia	6.0	2.888889	3

Colaborative Based

In [79]:

```
rating_utility_matrix = data.pivot_table(values ='overall_rating', index = 'i
rating_utility_matrix.head()
```

Out[79]:

airline_name	adria-airways	aegean-airlines	aer-lingus	aeroflot-russian-airlines	aerogal-aerolineas-galapagos	aerolineas-argentinas	aeromexico	aerosur
id_cus								
0	0.0	0.0	0.0	0.0	0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	0	0.0	0.0	0
2	0.0	0.0	0.0	0.0	0	0.0	0.0	0
3	0.0	0.0	2.0	0.0	0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0	0.0	0.0	0

5 rows × 292 columns

In [80]:

```
from sklearn.decomposition import TruncatedSVD
X = rating_utility_matrix.T
SVD = TruncatedSVD(n_components=30)
transposed_matrix = SVD.fit_transform(X)
```

In [81]:

```
corr_matrix = np.corrcoef(transposed_matrix)
airlines = rating_utility_matrix.columns
airline_list = list(airlines)
airline_index = airline_list.index('aegean-airlines')
airline_index
```

Out[81]:

```
1
```

In [82]:

```
type(airline_index)
```

Out[82]:

```
int
```

In [83]:

```
corr = corr_matrix[airline_index]
```

In [84]:

```
#Airlines which are more then 70% corellated
list(airlines[(corr < 1.0) & (corr > 0.7)])
```

Out[84]:

```
['aegean-airlines']
```

Hybrid Based

In [85]:

```
#Build the SVD based Collaborative filter
import surprise
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate
reader = surprise.Reader()
svd_data= Dataset.load_from_df(data[['id_cus', 'airline_id', 'overall_rating'
# Use the famous SVD algorithm.
svd = SVD()
# Run 5-fold cross-validation and print results.
cross_validate(svd, svd_data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    3.3303  3.3358  3.3316  3.3415  3.3688  3.3416  0.0142  
MAE (testset)     2.9699  2.9827  2.9838  2.9876  3.0054  2.9859  0.0114  
Fit time          1.46    1.45    1.46    1.46    1.48    1.46    0.01    
Test time         0.04    0.03    0.04    0.03    0.03    0.04    0.00    
```

Out[85]:

```
{'fit_time': (1.4567475318908691,
  1.4469928741455078,
  1.4582104682922363,
  1.4582836627960205,
  1.4794580936431885),
 'test_mae': array([2.96991946, 2.98273679, 2.98384251, 2.98759639, 3.00538
519]),
 'test_rmse': array([3.33028403, 3.33579736, 3.33164812, 3.34149318, 3.3688
2381]),
 'test_time': (0.03750038146972656,
  0.03420543670654297,
  0.04187178611755371,
  0.03348374366760254,
  0.033586978912353516)}
```

In [86]:

```
d = data['airline_name'].unique()
d = pd.DataFrame(d)
```

In [87]:

```
col = [0]
le= LabelEncoder()
d[1] = d[col].apply(le.fit_transform)
d
```

Out[87]:

	0	1
0	adria-airways	0
1	aegean-airlines	1
2	aer-lingus	2
3	aeroflot-russian-airlines	3
4	aerolineas-argentinas	5
...	...	...
287	wizz-air	287
288	wow-air	288
289	xiamen-airlines	289
290	xl-airways-france	290
291	yangon-airways	291

292 rows × 2 columns

In [88]:

```
d = d.set_index(0)
d = d[1]
d
```

Out[88]:

```
0
adria-airways                  0
aegean-airlines                1
aer-lingus                     2
aeroflot-russian-airlines      3
aerolineas-argentinas          5
                            ...
wizz-air                     287
wow-air                      288
xiamen-airlines              289
xl-airways-france            290
yangon-airways               291
Name: 1, Length: 292, dtype: int64
```

In [89]:
```
cosine_similarity_data =data.pivot_table(index='airline_id',columns='id_cus',
cosine_similarity_data.head()
```

Out[89]:

id_cus	0	1	2	3	4	5	6	7	8	9	...	19624	19625	19626	19627	1962
airline_id																
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0
2	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0

5 rows × 19634 columns

In [90]:
```
id_to_name = data[['airline_name', 'author', 'author_country', 'cabin_flown',
       'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
       'food_beverages_rating', 'inflight_entertainment_rating',
       'value_money_rating', 'Month', 'Year', 'recommended', 'id', 'Gender',
       'Customer Type', 'Age', 'Type of Travel', 'Flight Distance',
       'Inflight wifi service', 'Departure/Arrival time convenient',
       'Ease of Online booking', 'Gate location', 'Online boarding',
       'On-board service', 'Leg room service', 'Baggage handling',
       'Checkin service', 'Inflight service', 'Cleanliness', 'id_cus',
       'airline_id']]
```

In [91]:
```
id_to_name.drop_duplicates(subset ="id_cus",inplace = True)
```

In [92]:
```
id_to_name = id_to_name.set_index('id_cus')
```

In [93]:
```
name_to_id = data.set_index('airline_name')
```

In [94]:
```
name_to_id.drop_duplicates(subset ="airline_id",inplace = True)
```

In [95]:

```
Airlines = data[['airline_name', 'author', 'author_country', 'cabin_flown',
        'overall_rating', 'seat_comfort_rating', 'cabin_staff_rating',
        'food_beverages_rating', 'inflight_entertainment_rating',
        'value_money_rating', 'Month', 'Year', 'recommended', 'id', 'Gender',
        'Customer Type', 'Age', 'Type of Travel', 'Flight Distance',
        'Inflight wifi service', 'Departure/Arrival time convenient',
        'Ease of Online booking', 'Gate location', 'Online boarding',
        'On-board service', 'Leg room service', 'Baggage handling',
        'Checkin service', 'Inflight service', 'Cleanliness', 'id_cus',
        'airline_id']]
```

In [96]:

```
Airlines = Airlines.drop_duplicates(subset = 'airline_name',keep = 'first')
Airlines
```

Out[96]:

	airline_name	author	author_country	cabin_flown	overall_rating	seat_comfort_rati
0	adria-airways	D Ito	Germany	Economy	7.0	
17	aegean-airlines	Eric Botha	United Kingdom	Business Class	8.0	
32	aer-lingus	Keith Tynan	United States	Economy	3.0	
106	aeroflot-russian-airlines	A McAndrew	United States	Premium Economy	8.0	
152	aerolineas-argentinas	Hilarion Martinez	United States	Economy	5.0	
...	...	...	...	...	...	
26818	wizz-air	P Lako	Sweden	Economy	8.0	
27100	wow-air	Brian Seitz	United States	Economy	1.0	
27123	xiamen-airlines	Gunawanto Johannes Tamawidjaja	Indonesia	Economy	5.0	
27160	xl-airways-france	Christine Gayle	United States	Economy	5.0	
27223	yangon-airways	Jeff Nash	Australia	Economy	10.0	

292 rows × 32 columns

In [97]:

```
idx = d['aegean-airlines']
scores = list(enumerate(cosine_similarity_data[(int(idx))]))
```

In [98]:

```
scores = sorted(scores, key=lambda x: x[1], reverse=True)
```

In [99]:

```
airline_indices = [i[0] for i in scores]
```

In [100]:

```
airline_final = Airlines.iloc[airline_indices][['airline_name', 'overall_rati
```

In [101]:

```
airline_final['est'] = airline_final['id_cus'].apply(lambda x: svd.predict(1,
```

In [102]:

```
airline_final = airline_final.sort_values(['est'], ascending=False)
```

In [103]:

```
airline_final.head(5)
```

Out[103]:

	airline_name	overall_rating	id_cus	airline_id	est
20416	singapore-airlines	10.0	5776	233	5.0
17668	okay-airways	4.0	1129	195	5.0
16122	lufthansa	9.0	2668	176	5.0
16169	luxair	9.0	3769	177	5.0
16188	mahan-air	10.0	13182	178	5.0

In [104]:

```
airline_final = ((airline_final.head(10))['airline_name']).reset_index(drop=T
airline_final.index = airline_final.index +1
```

In [105]:

```
airline_final
```

Out[105]:

```
1      singapore-airlines
2            okay-airways
3               lufthansa
4                  luxair
5               mahan-air
6       malaysia-airlines
7           malm-aviation
8                   mango
9               meridiana
10          miat-mongolian
Name: airline_name, dtype: object
```

In [ ]: