In [34]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confus
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
```

In [35]:

```
df = pd.read_csv('airlines.csv')
```

In [36]:

```
df.shape
```

Out[36]:

```
(25976, 25)
```

In [37]:

```
df.head(10)
```

Out[37]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Depa time
0	0	19556	Female	Loyal Customer	52	Business travel	Eco	160	5	
1	1	90035	Female	Loyal Customer	36	Business travel	Business	2863	1	
2	2	12360	Male	disloyal Customer	20	Business travel	Eco	192	2	
3	3	77959	Male	Loyal Customer	44	Business travel	Business	3377	0	
4	4	36875	Female	Loyal Customer	49	Business travel	Eco	1182	2	
5	5	39177	Male	Loyal Customer	16	Business travel	Eco	311	3	
6	6	79433	Female	Loyal Customer	77	Business travel	Business	3987	5	
7	7	97286	Female	Loyal Customer	43	Business travel	Business	2556	2	
8	8	27508	Male	Loyal Customer	47	Business travel	Eco	556	5	
9	9	62482	Female	Loyal Customer	46	Business travel	Business	1744	2	

10 rows × 25 columns

In [38]:

```
df.tail(10)
```

Out[38]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service
25966	25966	30263	Male	disloyal Customer	42	Business travel	Eco	1024	4
25967	25967	90347	Female	disloyal Customer	39	Business travel	Business	404	1
25968	25968	86816	Male	Loyal Customer	41	Business travel	Eco	692	2
25969	25969	120654	Male	Loyal Customer	52	Business travel	Business	280	3
25970	25970	25309	Female	disloyal Customer	36	Business travel	Eco	432	1
25971	25971	78463	Male	disloyal Customer	34	Business travel	Business	526	3
25972	25972	71167	Male	Loyal Customer	23	Business travel	Business	646	4
25973	25973	37675	Female	Loyal Customer	17	Personal Travel	Eco	828	2
25974	25974	90086	Male	Loyal Customer	14	Business travel	Business	1127	3
25975	25975	34799	Female	Loyal Customer	42	Personal Travel	Eco	264	2

10 rows × 25 columns

In [39]:
```
df.dtypes
```

Out[39]:
```
Unnamed: 0                             int64
id                                     int64
Gender                                object
Customer Type                         object
Age                                    int64
Type of Travel                        object
Class                                 object
Flight Distance                        int64
Inflight wifi service                  int64
Departure/Arrival time convenient      int64
Ease of Online booking                 int64
Gate location                          int64
Food and drink                         int64
Online boarding                        int64
Seat comfort                           int64
Inflight entertainment                 int64
On-board service                       int64
Leg room service                       int64
Baggage handling                       int64
Checkin service                        int64
Inflight service                       int64
Cleanliness                            int64
Departure Delay in Minutes             int64
Arrival Delay in Minutes             float64
satisfaction                          object
dtype: object
```

In [40]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25976 entries, 0 to 25975
Data columns (total 25 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Unnamed: 0                         25976 non-null  int64  
 1   id                                 25976 non-null  int64  
 2   Gender                             25976 non-null  object 
 3   Customer Type                      25976 non-null  object 
 4   Age                                25976 non-null  int64  
 5   Type of Travel                     25976 non-null  object 
 6   Class                              25976 non-null  object 
 7   Flight Distance                    25976 non-null  int64  
 8   Inflight wifi service              25976 non-null  int64  
 9   Departure/Arrival time convenient  25976 non-null  int64  
 10  Ease of Online booking             25976 non-null  int64  
 11  Gate location                      25976 non-null  int64  
 12  Food and drink                     25976 non-null  int64  
 13  Online boarding                    25976 non-null  int64  
 14  Seat comfort                       25976 non-null  int64  
 15  Inflight entertainment             25976 non-null  int64  
 16  On-board service                   25976 non-null  int64  
 17  Leg room service                   25976 non-null  int64  
 18  Baggage handling                   25976 non-null  int64  
 19  Checkin service                    25976 non-null  int64  
 20  Inflight service                   25976 non-null  int64  
 21  Cleanliness                        25976 non-null  int64  
 22  Departure Delay in Minutes         25976 non-null  int64  
 23  Arrival Delay in Minutes           25893 non-null  float64
 24  satisfaction                       25976 non-null  object 
dtypes: float64(1), int64(19), object(5)
memory usage: 5.0+ MB
```

In [41]:

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

Out[41]:

```
Unnamed: 0                            0
id                                    0
Gender                                0
Customer Type                         0
Age                                   0
Type of Travel                        0
Class                                 0
Flight Distance                       0
Inflight wifi service                 0
Departure/Arrival time convenient     0
Ease of Online booking                0
Gate location                         0
Food and drink                        0
Online boarding                       0
Seat comfort                          0
Inflight entertainment                0
On-board service                      0
Leg room service                      0
Baggage handling                      0
Checkin service                       0
Inflight service                      0
Cleanliness                           0
Departure Delay in Minutes            0
Arrival Delay in Minutes             83
satisfaction                          0
dtype: int64
```

In [42]:

```
df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].mean(
```

Analyzing the data

In [43]:

```
df.describe()
```

Out[43]:

	Unnamed: 0	id	Age	Flight Distance	Inflight wifi service	Departure/A time conv
count	25976.000000	25976.000000	25976.000000	25976.000000	25976.000000	25976.0
mean	12987.500000	65005.657992	39.620958	1193.788459	2.724746	3.0
std	7498.769632	37611.526647	15.135685	998.683999	1.335384	1.5
min	0.000000	17.000000	7.000000	31.000000	0.000000	0.0
25%	6493.750000	32170.500000	27.000000	414.000000	2.000000	2.0
50%	12987.500000	65319.500000	40.000000	849.000000	3.000000	3.0
75%	19481.250000	97584.250000	51.000000	1744.000000	4.000000	4.0
max	25975.000000	129877.000000	85.000000	4983.000000	5.000000	5.0

Data preprocessing

In [44]:

```
df2=df.drop(['Unnamed: 0','id','Customer Type','Type of Travel','Gate loca
```

In [45]:

```
df2
```

Out[45]:

	Gender	Age	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Food and drink	Onli boardi
0	Female	52	Eco	160	5	4	3	3	
1	Female	36	Business	2863	1	1	3	5	
2	Male	20	Eco	192	2	0	2	2	
3	Male	44	Business	3377	0	0	0	3	
4	Female	49	Eco	1182	2	3	4	4	
...	...	...	...	...	...	...	...	...	
25971	Male	34	Business	526	3	3	3	4	
25972	Male	23	Business	646	4	4	4	4	
25973	Female	17	Eco	828	2	5	1	2	
25974	Male	14	Business	1127	3	3	3	4	
25975	Female	42	Eco	264	2	5	2	4	

25976 rows × 20 columns

In [46]:

```
for x in df2.select_dtypes(include = 'object'):
    print(df2[x].value_counts())
```

```
Female    13172
Male      12804
Name: Gender, dtype: int64
Business    12495
Eco         11564
Eco Plus     1917
Name: Class, dtype: int64
neutral or dissatisfied    14573
satisfied                  11403
Name: satisfaction, dtype: int64
```

In [47]:

```
df2['Gender'] = df2['Gender'].replace({'Male': 1, 'Female': 0})
df2['satisfaction'] = df2['satisfaction'].replace({'neutral or dissatisfie
df2['Class'] = df2['Class'].replace({'Business': 0, 'Eco': 1,'Eco Plus': 2
In [66]: df2.head(10)
```

Out[66]:

	Gender	Age	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Food and drink	Online boarding	con
0	0	52	1	160	5	4	3	3	4	
1	0	36	0	2863	1	1	3	5	4	
2	1	20	1	192	2	0	2	2	2	
3	1	44	0	3377	0	0	0	3	4	
4	0	49	1	1182	2	3	4	4	1	
5	1	16	1	311	3	3	3	5	5	
6	0	77	0	3987	5	5	5	3	5	
7	0	43	0	2556	2	2	2	4	4	
8	1	47	1	556	5	2	2	5	5	
9	0	46	0	1744	2	2	2	3	4	

```
In [68]: df2.tail(10)
```

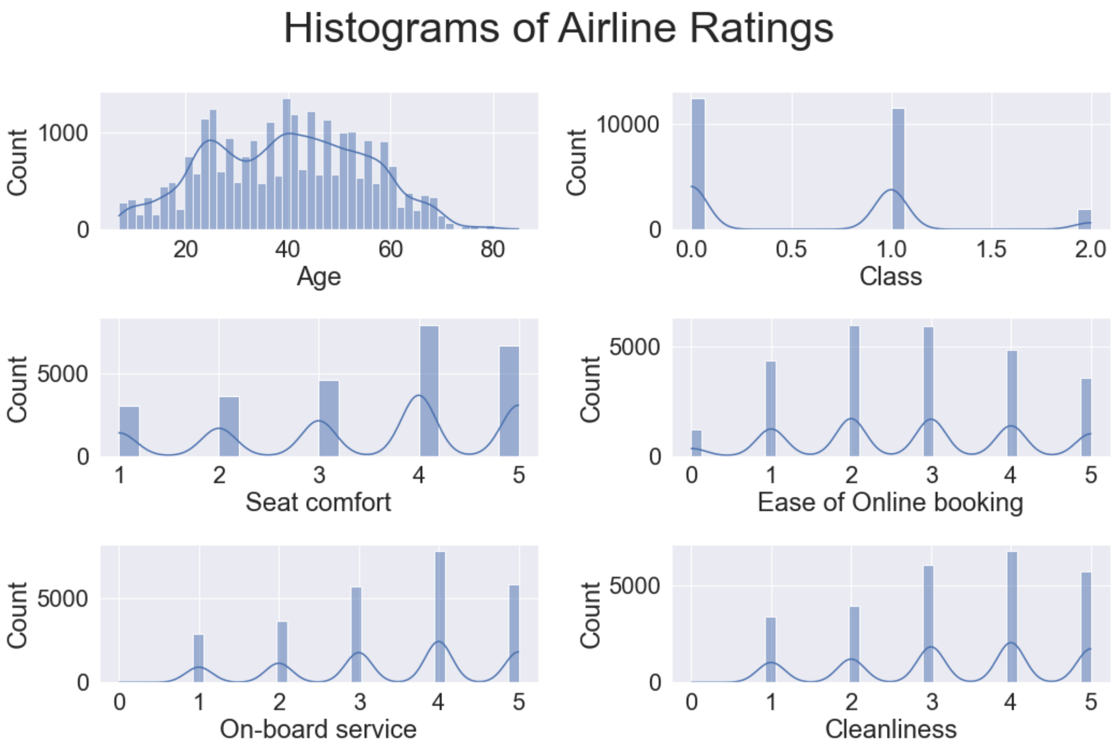
Out[68]:

	Gender	Age	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Food and drink	Online boarding
25966	1	42	1	1024	4	4	4	3	4
25967	0	39	0	404	1	1	1	2	1
25968	1	41	1	692	2	2	2	2	2
25969	1	52	0	280	3	3	3	3	4
25970	0	36	1	432	1	5	1	4	1
25971	1	34	0	526	3	3	3	4	3
25972	1	23	0	646	4	4	4	4	4
25973	0	17	1	828	2	5	1	2	1
25974	1	14	0	1127	3	3	3	4	4
25975	0	42	1	264	2	5	2	4	2

visualizing the data

In [77]:

```
sns.set_style('darkgrid')
# createing a 3X2 subplots
#This adds a Kernel Density Estimate (KDE)
#plot to the histogram, providing a smoothed representation of the distrib

fig, axs = plt.subplots (nrows=3, ncols=2, figsize=(15, 10))
sns.histplot(ax=axs[0, 0], data=df2, x='Age', kde=True)
sns.histplot(ax=axs[0, 1], data=df2, x='Class', kde=True)
sns.histplot(ax=axs[1, 0], data=df2, x='Seat comfort', kde=True)
sns.histplot(ax=axs[1, 1], data=df2, x='Ease of Online booking', kde=True)
sns.histplot(ax=axs[2, 0], data=df2, x='On-board service', kde=True)
sns.histplot(ax=axs[2, 1], data=df2, x='Cleanliness', kde=True)
# Adding title
fig.suptitle('Histograms of Airline Ratings', fontsize=40)
plt.tight_layout()
plt.show()
```



The ways of styling themes are as follows:

white

dark

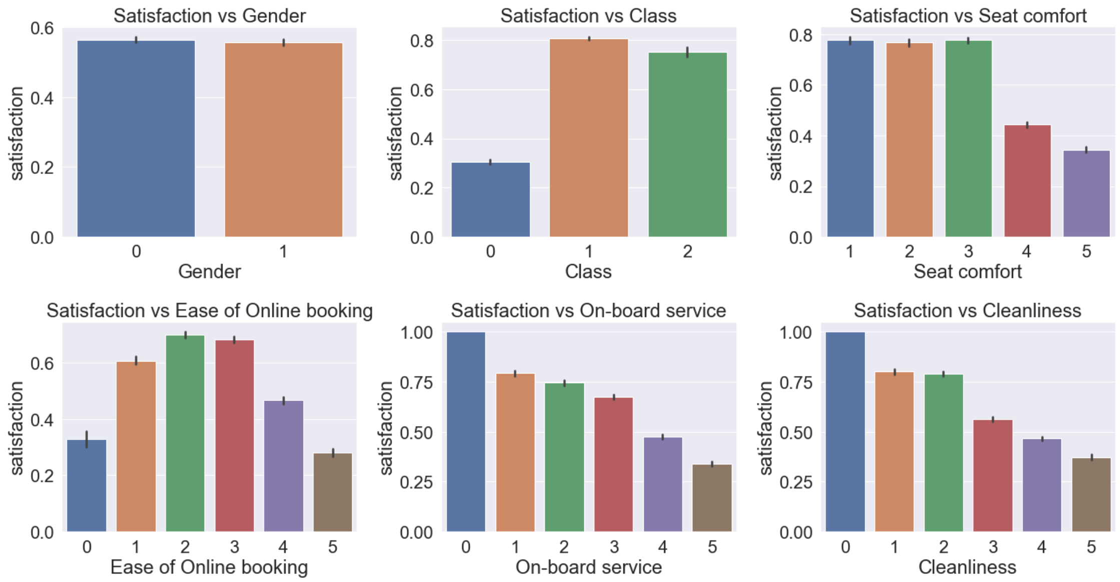
whitegrid

darkgrid

ticks

In [78]:

```
plt.figure(figsize=(20, 15))

# Set the font scale
sns.set(font_scale=2)

# Create a 3x2 grid of bar plots
plt.subplot(3, 3, 1)
sns.barplot(y='satisfaction', x='Gender', data=df2)
plt.title("Satisfaction vs Gender")

plt.subplot(3, 3, 2)
sns.barplot(y='satisfaction', x='Class', data=df2)
plt.title("Satisfaction vs Class")

plt.subplot(3, 3, 3)
sns.barplot(y='satisfaction', x='Seat comfort', data=df2)
plt.title("Satisfaction vs Seat comfort")

plt.subplot(3, 3, 4)
sns.barplot(y='satisfaction', x='Ease of Online booking', data=df2)
plt.title("Satisfaction vs Ease of Online booking")

plt.subplot(3, 3, 5)
sns.barplot(y='satisfaction', x='On-board service', data=df2)
plt.title("Satisfaction vs On-board service")

plt.subplot(3, 3, 6)
sns.barplot(y='satisfaction', x='Cleanliness', data=df2)
plt.title("Satisfaction vs Cleanliness")

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```



In [52]:

```
df2[["Age","satisfaction"]].groupby(["Age"],as_index=False).mean().sort_va
```

Out[52]:

	Age	satisfaction
0	7	0.926829
1	8	0.923567
63	70	0.906040
62	69	0.882353
4	11	0.867925
...	...	...
49	56	0.393805
39	46	0.386121
34	41	0.385466
46	53	0.382664
44	51	0.369781

75 rows × 2 columns

In [53]:

```
df2[["Gender","satisfaction"]].groupby(["Gender"],as_index=False).mean().s
```

Out[53]:

	Gender	satisfaction
0	0	0.564607
1	1	0.557326

Training the data

In [54]:

```
y=df2['satisfaction']
x=df2.drop('satisfaction',axis=1)
```

In [55]:

```
print(x)
```

```
       Gender  Age  Class  Flight Distance  Inflight wifi service  \
0           0   52      1              160                      5
1           0   36      0             2863                      1
2           1   20      1              192                      2
3           1   44      0             3377                      0
4           0   49      1             1182                      2
...       ...  ...    ...              ...                    ...
25971       1   34      0              526                      3
25972       1   23      0              646                      4
25973       0   17      1              828                      2
25974       1   14      0             1127                      3
25975       0   42      1              264                      2

       Departure/Arrival time convenient  Ease of Online booking  \
0                                      4                       3
1                                      1                       3
2                                      0                       2
3                                      0                       0
4                                      3                       4
```

In [56]:

```
print(y)
```

```
0        0
1        0
2        1
3        0
4        0
        ..
25971    1
25972    0
25973    1
25974    0
25975    1
Name: satisfaction, Length: 25976, dtype: int64
```

In [57]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.25
```

In [58]:

```
print(x.shape,x_test.shape,x_train.shape)
```

```
(25976, 19) (6494, 19) (19482, 19)
```

In [59]:

```
print(y.shape,y_test.shape,y_train.shape)
```

```
(25976,) (6494,) (19482,)
```

In [60]:

```
model_accuracy=pd.DataFrame(columns=['model','Accuracy'])
```

Random Forest

In [61]:

```
model =  RandomForestClassifier()
```

In [62]:

```
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
```

In [63]:

```
# Utilizing testing set to test model accuracy = model.score(x_test, y_tes

print('RandomForestClassifier')
print(f'Model_Accuracy\t\t: {accuracy}')
print(f'Accuracy in Percentage\t: {"{:.1%}".format(accuracy)}')
print(classification_report(y_test, y_pred))
```

```
RandomForestClassifier
Model_Accuracy          : 0.9430243301509086
Accuracy in Percentage  : 94.3%
              precision    recall  f1-score   support

           0       0.95      0.92      0.94      2904
           1       0.94      0.96      0.95      3590

    accuracy                           0.94      6494
   macro avg       0.94      0.94      0.94      6494
weighted avg       0.94      0.94      0.94      6494
```

Precision is the ratio of correctly predicted positive observations to the total predicted positives.

Recall is the ratio of correctly predicted positive observations to the all observations in actual class

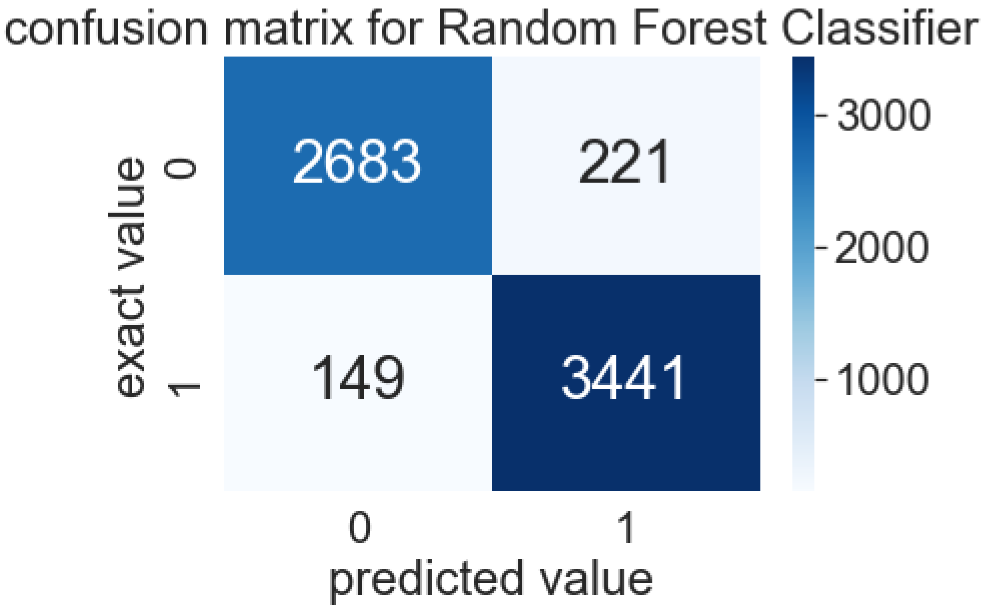
The F1-score is the weighted average of precision and recall. It ranges from 0 to 1, where 1 is the best possible F1-score.

The number of actual occurrences of the class in the specified dataset. For class 0, there are 2904 instances, and for class 1, there are 3590 instances.

The overall accuracy of the model is 94.3%, which is the ratio of correctly predicted instances to the total instances.

In [64]:

```
# confusion matrix for random forest

con_matrrix = confusion_matrix(y_test, y_pred)

# Create a heatmap of the confusion matrix

sns.set(font_scale=2) # Set font size

sns.heatmap(con_matrrix, annot=True, annot_kws={"size": 30}, cmap='Blues'
plt.xlabel('predicted value')
plt.ylabel('exact value')
plt.title('confusion matrix for Random Forest Classifier')
plt.show()
```


confusion matrix for Random Forest Classifier
exact value
predicted value