Airline Passenger Satisfaction
What factors lead to customer satisfaction for an Airline?

Python Data Analysis Project
Jose V. Cuan
January 30, 2022
NYC Data Science Academy

Introduction

For my Python Data Analysis Project, I decided to use the dataset "Airline Passenger Satisfaction" (APS) from Kaggle (https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction) [1].

This dataset contains an airline passenger satisfaction survey (statistical data) which we will use to apply different python commands/techniques learned in class.

In this contest, we can formulate some research questions:

- What factors are highly correlated to a satisfied (or dissatisfied) passenger?
- Can you predict passenger satisfaction?
- Which Model can you use to predict APS and Why?

Data observation

Before getting data preparation and classification of the dataset (train and test file), let's evaluate what kind of data we have.

Content data:

- Gender: Gender of the passengers (Female, Male)
- Customer Type: The customer type (Loyal customer, disloyal customer)
- Age: The actual age of the passengers
- Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)
- Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)
- Flight distance: The flight distance of this journey
- Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)
- Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient
- Ease of Online booking: Satisfaction level of online booking
- Gate location: Satisfaction level of Gate location
- Food and drink: Satisfaction level of Food and drink
- Online boarding: Satisfaction level of online boarding
- Seat comfort: Satisfaction level of Seat comfort
- Inflight entertainment: Satisfaction level of inflight entertainment

- On-board service: Satisfaction level of On-board service
- Leg room service: Satisfaction level of Leg room service
- Baggage handling: Satisfaction level of baggage handling
- Check-in service: Satisfaction level of Check-in service
- Inflight service: Satisfaction level of inflight service
- Cleanliness: Satisfaction level of Cleanliness
- Departure Delay in Minutes: Minutes delayed when departure
- Arrival Delay in Minutes: Minutes delayed when Arrival
- Satisfaction: Airline satisfaction level (Satisfaction, neutral or dissatisfaction)

[ ] train_APS.head() #train dataframe

Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	D
Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	4	5	5	
Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	1	4	1	
Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4	3	4	4	4	5	
Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2	5	3	1	4	2	
Male	Loyal Customer	61	Business travel	Business	214	3	3	...	3	3	4	4	3	3	3	

Figure 1. Train Dataframe

[ ] train_APS.nunique() # Observation of what Type of Data (Categorical or Numerical) is in the Dataset.

Table 1: Type of Data (Categorical vs Numerical)

id	103904
Gender	2
Customer Type	2
Age	75
Type of Travel	2
Class	3
Flight Distance	3802
Inflight wifi service	6
Departure/Arrival time convenient	6
Ease of Online booking	6
Gate location	6
Food and drink	6
Online boarding	6
Seat comfort	6
Inflight entertainment	6
On-board service	6
Leg room service	6
Baggage handling	5
Checkin service	6
Inflight service	6
Cleanliness	6
Departure Delay in Minutes	446

Arrival Delay in Minutes	455
satisfaction	2

From this list of variables, we can classify the dataset as follow:

- Categorical: "Gender", "Customer Type", "Type of Travel", "Class" and "Satisfaction".
- Numerical: "Age", "Flight Distance", "Departure Delay" and "Arrival Delay".
- Grade (1-5): Inflight wifi service, Departure/Arrival time convenient, Ease of Online booking, Gate location, Food and drink, Online boarding, Seat comfort, Inflight entertainment, On-board service, Leg room service, Baggage handling, Checkin service, Inflight service, and Cleanliness.

Data preparation

From this dataset, we found that only "Arrival Delay in Minutes" has missing values. This variable has just less than 0.3% of missing values. We can see this minimal percentage amount in Figure 2.

```
[ ] train_APS.isnull().sum()*100/len(train_APS) # Found less than 0.3% missing values on "Arrival Delay"
[ ] msno.matrix(train_APS) # heatmap to visualize missing data in a dataframe
```

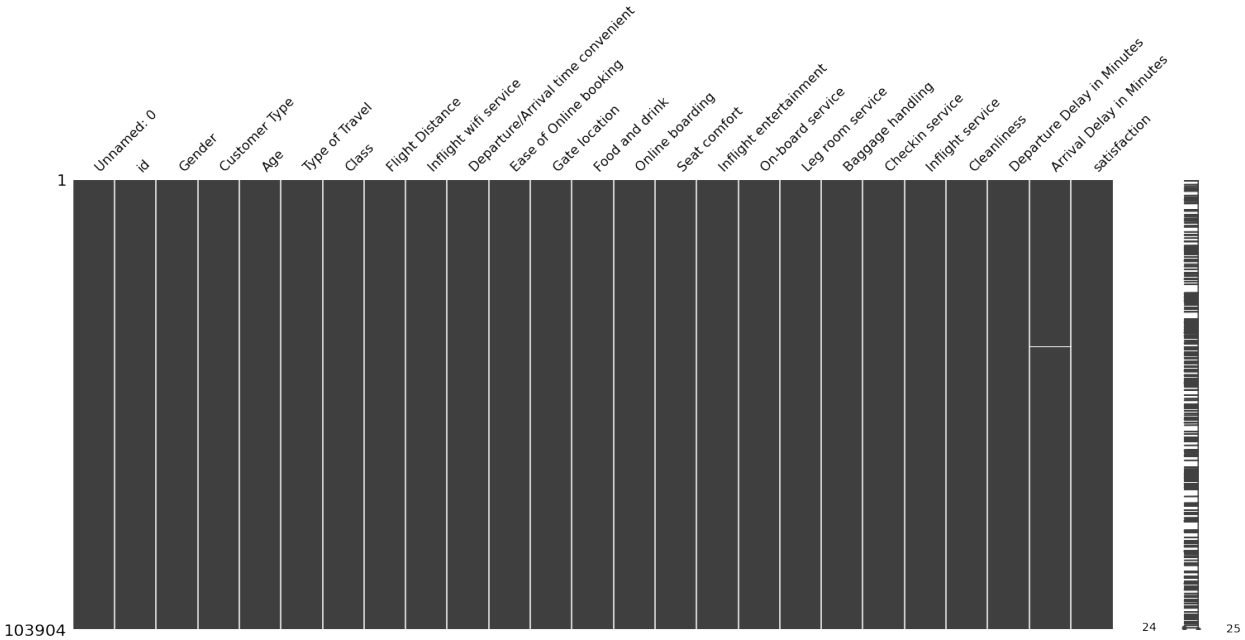


Figure 2. Less than 0.3% of missing values in 'Arrival Delay in Minute" variable.

We are going to eliminate (drop) those rows in which missing values are present.

```
[ ] train_APS.dropna(subset = ['Arrival Delay in Minutes'], inplace=True)
```

Data Analysis

At this point, we are ready to do some Data Analysis. Let's start with visualizing the correlation between variables in a heatmap. Also, we need to encode categorical data into numerical data.

Table 2: Label Encoding Data

Variable	Label Encoding
Gender	1 (Male)
	0 (Female)
Customer Type	1 (Loyal Customer)
	0 (Disloyal Customer)
Type of Travel	1 (Business travel)
	0 (Personal travel)
Class	2 (Business)
	1 (Eco Plus)
	0 (Eco)
Satisfaction	1 (Satisfied)
	0 (dissatisfied)

```
[ ] train_APS['Gender'].replace(['Female', 'Male'], [0, 1], inplace=True)
[ ] train_APS['Customer Type'].replace(['disloyal Customer', 'Loyal Customer'], [0, 1], inplace=True)
[ ] train_APS['Type of Travel'].replace(['Personal Travel', 'Business travel'], [0, 1], inplace=True)
[ ] train_APS['Class'].replace(['Eco', 'Eco Plus', 'Business'], [0, 1, 2], inplace=True)
[ ] train_APS['satisfaction'].replace(['neutral or dissatisfied', 'satisfied'], [0, 1], inplace=True)
```

After this process, let's see what our dataframe look like:

```
[ ] train_APS.head(10)
```

Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	Departu Delay Minut
1	1	13	0	1	460	3	4	...	5	4	3	4	4	5	5	
1	0	25	1	2	235	3	2	...	1	1	5	3	1	4	1	
0	1	26	1	2	1142	2	2	...	5	4	3	4	4	4	5	
0	1	25	1	2	562	2	5	...	2	2	5	3	1	4	2	
1	1	61	1	2	214	3	3	...	3	3	4	4	3	3	3	
0	1	26	0	0	1180	3	4	...	1	3	4	4	4	4	1	
1	1	47	0	0	1276	2	4	...	2	3	3	4	3	5	2	
0	1	52	1	2	2035	4	3	...	5	5	5	5	4	5	4	
0	1	41	1	2	853	1	2	...	1	1	2	1	4	1	2	
1	0	20	1	0	1061	3	3	...	2	2	3	4	4	3	2	

Figure 3. Test Dataframe with categorical data encoded

And finally, the correlation heatmap of our dataframe:

```
[ ] sns.heatmap(train_APS.corr(), cmap='YlGnBu', xticklabels=True, yticklabels=True)
```

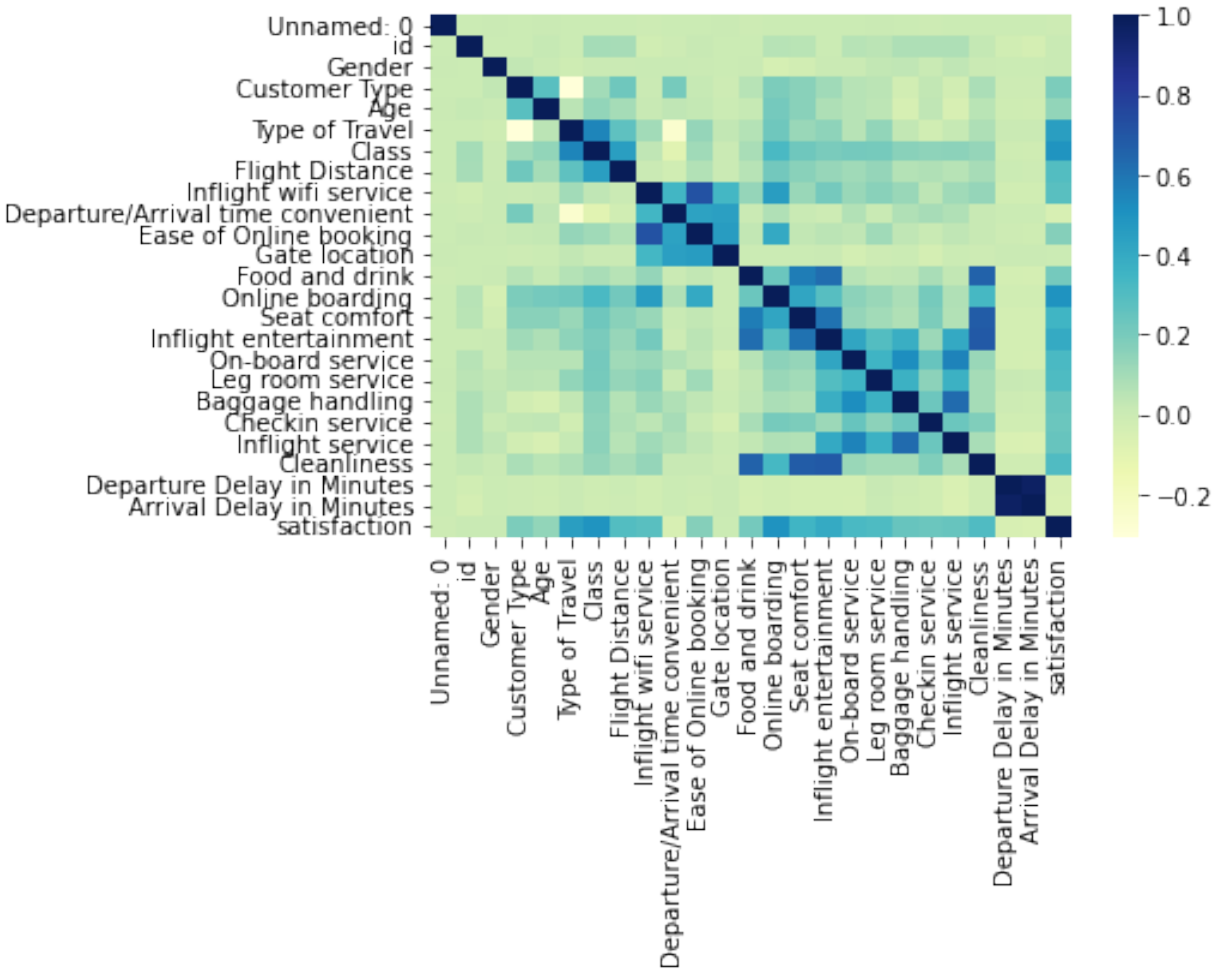


Figure 4. Correlation heatmap of variables.

From the correlation heatmap, we need to numerically identify which variables are highly correlated and use them for our prediction model. In the heatmap, the rightmost column shows the correlation between features and target (satisfaction).

```
[ ] corr = train_APS.corr()
[ ] Rank_Corr_List = corr[corr<1].unstack().transpose().sort_values(ascending=False).drop_duplicates()
[ ] Rank_Corr_List.head(20)
```

Table 3: Correlation index between variables.

Arrival Delay in Minutes	Departure Delay in Minutes	0.965481
Ease of Online booking	Inflight wifi service	0.715848
Cleanliness	Inflight entertainment	0.691735
	Seat comfort	0.678478
Food and drink	Cleanliness	0.657648
Baggage handling	Inflight service	0.628944
Inflight entertainment	Food and drink	0.622374
	Seat comfort	0.610614
Seat comfort	Food and drink	0.574561
On-board service	Inflight service	0.550725

Type of Travel	Class	0.545185
On-board service	Baggage handling	0.519252
Online boarding	satisfaction	0.503447
Class	satisfaction	0.494545
Ease of Online booking	Gate location	0.458746
Online boarding	Inflight wifi service	0.457002
Flight Distance	Class	0.451495
Type of Travel	satisfaction	0.448995
Departure/Arrival time convenient	Gate location	0.444601
	Ease of Online booking	0.437021

We are going to consider, just the highest correlated variables (correlation > 0.45) for our prediction model (including "satisfaction"). We are going to eliminated (drop) from our dataset those variables/columns that do not have a significant correlation.

```
[ ] train_APS.drop(['Gender', 'Age', 'Flight Distance', 'Departure/Arrival time convenient', 'Ease of Online booking',
'On-board service', 'Leg room service', 'Inflight service', 'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay
in Minutes'], axis=1, inplace=True)
[] train_APS.head()
```

After eliminating low correlated variables/columns, our dataset table looks as follow:

	Unnamed: 0	id	Customer Type	Type of Travel	Class	Inflight wifi service	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	Baggage handling	Checkin service	satisfaction
0	0	70172	1	0	1	3	1	5	3	5	5	4	4	0
1	1	5047	0	1	2	3	3	1	3	1	1	3	1	0
2	2	110028	1	1	2	2	2	5	5	5	5	4	4	1
3	3	24026	1	1	2	2	5	2	2	2	2	3	1	0
4	4	119299	1	1	2	3	3	4	5	5	3	4	3	1

Figure 5. Train Dataframe after drop low correlated variables

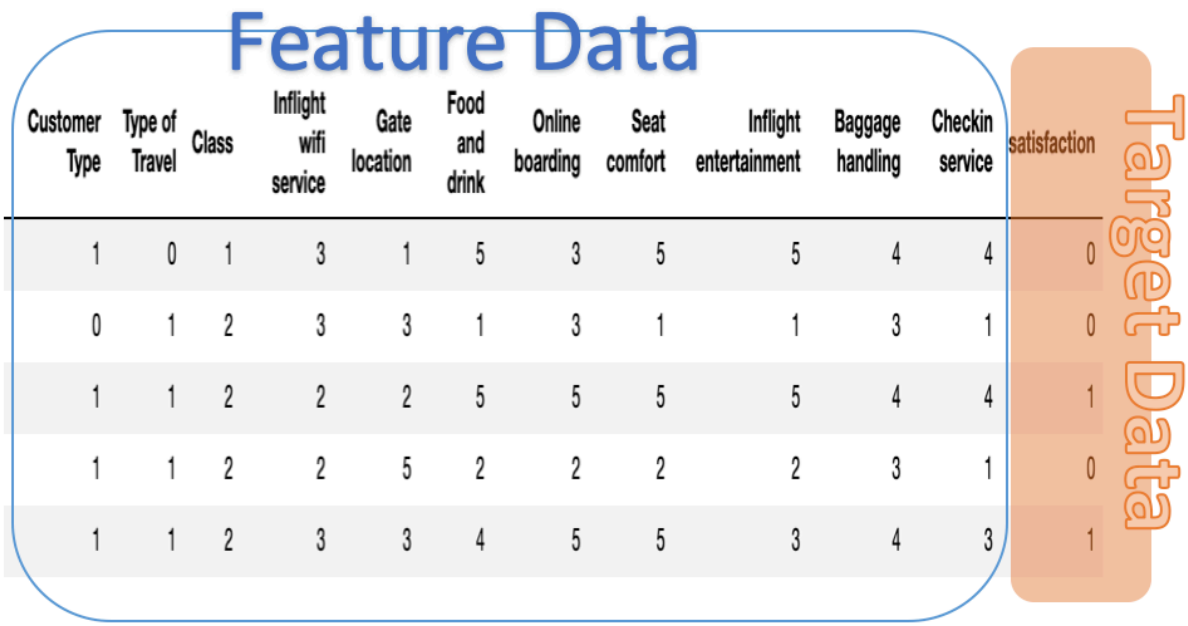
Building the Prediction Model

In order to get ready to build our classification prediction model, using the Random Forest technique, we need to separate the data into the “Features” and “Targets” data (Figure 6).

The target data, also known as the “label”, is the value we want to predict, in this case, the “satisfaction” of APS, and the features are all the variables/columns we select from the correlation heatmap (variables with high correlation).

We will also convert the Pandas-dataframes to Numpy-arrays, in order to make those variables suitable for the algorithm:

```
[ ] train_labels = np.array(train_APS['satisfaction']) #Label or target data
[ ] train_APS = train_APS.drop('satisfaction', axis = 1) #Remove label from DF, to become feature data.
[ ] train_APS = np.array(train_APS)
```



Customer Type	Type of Travel	Class	Inflight wifi service	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	Baggage handling	Checkin service	satisfaction
1	0	1	3	1	5	3	5	5	4	4	0
0	1	2	3	3	1	3	1	1	3	1	0
1	1	2	2	2	5	5	5	5	4	4	1
1	1	2	2	5	2	2	2	2	3	1	0
1	1	2	3	3	4	5	5	3	4	3	1

Figure 6. Feature Data vs Target (Label) Data to train our model.

Training the Model:

Now that the variables have been converted into arrays, we are ready to apply the sklearn library to train our model.

```
[ ] from sklearn.model_selection import train_test_split
[ ] from sklearn.ensemble import RandomForestClassifier
[ ] from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
[ ] %matplotlib inline

[ ] rf = RandomForestClassifier()
[ ] rf.fit(train_APS, train_labels)
```

Testing the Model:

The classification Random Forest Model is already trained; let's test it to see how accurate it is. We also need to adequate the test dataset (test_APS) in the same way we did with the train_APS dataset.

```
[ ] test_APS.dropna(subset = ['Arrival Delay in Minutes'], inplace=True)
[ ] test_APS['satisfaction'].replace(['neutral or dissatisfied', 'satisfied'], [0, 1], inplace=True)
[ ] test_APS['Customer Type'].replace(['disloyal Customer', 'Loyal Customer'], [0, 1], inplace=True)
[ ] test_APS['Type of Travel'].replace(['Personal Travel', 'Business travel'], [0, 1], inplace=True)
[ ] test_APS['Class'].replace(['Eco', 'Eco Plus', 'Business'], [0, 1, 2], inplace=True)
```

```
[ ] test_APS.drop(['Gender', 'Age', 'Flight Distance', 'Departure/Arrival time convenient', 'Ease of Online booking',
'On-board service', 'Leg room service', 'Inflight service', 'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay
in Minutes'], axis=1, inplace=True)

[ ] test_labels = np.array(test_APS['satisfaction']) # Test Label
[ ] test_APS = test_APS.drop('satisfaction', axis = 1) #Remove label from DF
[ ] test_APS = np.array(test_APS)
[ ] predictions = rf.predict(test_APS) #Use RF predict on test data
```

Let's see how accurate our model is:

```
[ ] accuracy_score(test_labels, predictions)

0.9578264395782644
```

95.78% accurate is pretty good for a Random Forest classification model.

Finally, let's get a graphic representation of our classification prediction model; "The Confusion Matrix"

```
[ ] matrix = confusion_matrix(test_labels, predictions)
[ ] matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]
[ ] plt.figure(figsize=(16,7))
[ ] sns.set(font_scale=1.4)
[ ] sns.heatmap(matrix, annot=True, annot_kws={'size':10},
        cmap=plt.cm.Greens, linewidths=0.2)

[ ] plt.xlabel('Predicted label')
[ ] plt.ylabel('True label')
[ ] plt.title('Confusion Matrix for Random Forest Model')
[ ] plt.show()
```

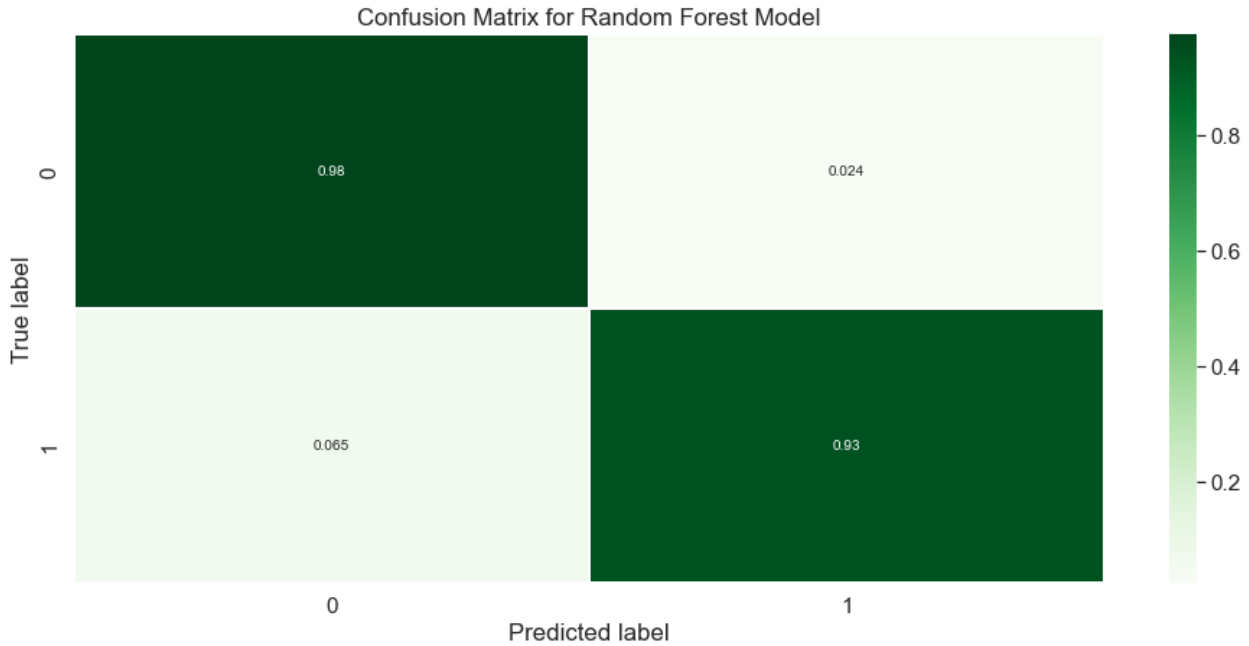


Figure 7. Confusion Matrix for Random Forest Model

Reference

1. https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction