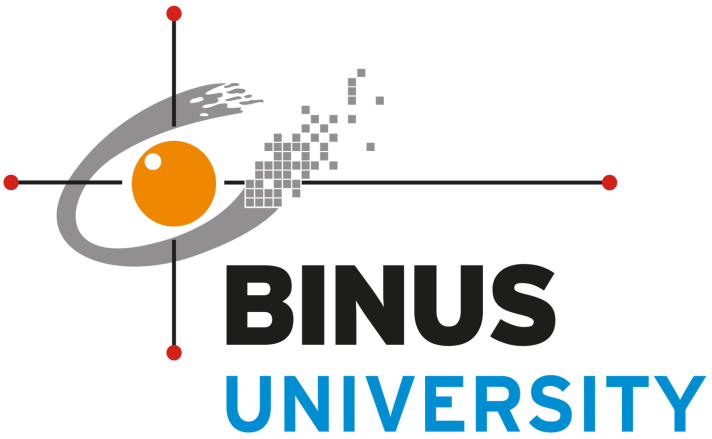
LAPORAN AOL MACHINE LEARNING

"AIRPLANE SATISFACTION"

MATA KULIAH MACHINE LEARNING LA09



Oleh:

Raiyen Dewi Kusuma	**2540118725**
Rafael Nicholas Tanaja	**2540118656**
Kevina Nugraha Eleeas	**2540120585**

School Of Computer Science

Universitas Bina Nusantara

Jakarta

2022

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan zaman, manusia semakin dipermudah untuk melakukan berbagai aktivitas dan pekerjaannya. Manusia dipermudah dalam melakukan perjalanan jarak jauh dengan banyaknya pilihan transportasi saat ini, seperti motor, mobil, kereta api, kapal, dan pesawat. Perjalanan antar pulau yang dulunya ditempuh menggunakan kapal selama berhari-hari, saat ini dapat dilakukan dengan mudah dan cepat menggunakan pesawat. Pesawat merupakan transportasi udara yang memungkinkan seseorang melakukan perjalanan jarak jauh antar benua dapat dilakukan dalam hitungan jam.

Saat ini terdapat banyak jenis maskapai penerbangan, tentunya dengan harga dan kualitas yang bervariasi menyesuaikan dengan pelanggannya. Pada umumnya harga tiket pesawat dipengaruhi oleh faktor jarak yang ditempuh, jenis maskapai penerbangan, dan kelas tiket pesawat yang akan mempengaruhi pelayanan selama penerbangan. Dengan banyaknya pilihan maskapai dan perbedaan harga seringkali membuat pembeli bingung untuk menentukan pesawat mana yang akan digunakan untuk menempuh perjalanan jauh agar nyaman dan tidak merasa lelah. Oleh karena itu, kami membuat model untuk memprediksikan kepuasan pelanggan berdasarkan beberapa faktor menggunakan metode machine learning. Kepuasan pelanggan menjadi tolak ukur untuk menentukan kualitas sebuah maskapai, sehingga pembeli dapat menemukan maskapai yang tepat untuk menemani perjalanannya.

1.2 Tujuan

Penelitian ini bertujuan untuk membuat model mengenai kepuasan pelanggan maskapai penerbangan dengan menggunakan metode dan teknik machine learning. Penelitian ini juga bertujuan untuk memberikan evaluasi terhadap maskapai penerbangan agar dapat mengetahui faktor-faktor yang perlu di kembangkan kembali, sehingga dapat meningkatkan kepuasan, kepercayaan, dan loyalitas pelanggan maskapai.

BAB II
METODE PENELITIAN

Penelitian ini menggunakan *dataset* yang didapat melalui *kaggle*, yang memiliki 2 bagian data yaitu *train set* dan *test set*. Penelitian ini juga menggunakan berbagai metode yang terdapat didalam *machine learning*.

2.1 *Data Overview*

Terdapat 5 teknik dalam menganalisa data yang digunakan dalam penelitian ini antara lain:

a. .read_csv

Metode ini untuk membaca data yang berbentuk csv, fungsi ini menggunakan

```
In [2]: # membuat copy dataframe `df_eda` untuk digunakan pada saat proses EDA.
        df_ori = pd.read_csv("airplanetrain.csv")
        df_eda = df_ori.copy()
```

read_csv dari *library pandas*.

b. .head

Metode ini digunakan untuk menampilkan sepuluh data teratas didalam *dataset*.

```
In [3]: df_eda.head()
```

Out[3]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arr time conveni
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	

c. *.info*

Metode ini untuk melihat nama kolom, jumlah kolom yang memiliki nilai atau tidak *null*, dan *data type* tiap kolom digunakan fungsi *info*.

```
In [4]: # memeriksa datatype setiap kolom yang ada di dataframe.
        df_eda.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Unnamed: 0                        103904 non-null  int64
 1   id                                103904 non-null  int64
 2   Gender                            103904 non-null  object
 3   Customer Type                     103904 non-null  object
 4   Age                               103904 non-null  int64
 5   Type of Travel                    103904 non-null  object
 6   Class                             103904 non-null  object
 7   Flight Distance                   103904 non-null  int64
 8   Inflight wifi service             103904 non-null  int64
 9   Departure/Arrival time convenient 103904 non-null  int64
 10  Ease of Online booking            103904 non-null  int64
 11  Gate location                     103904 non-null  int64
 12  Food and drink                    103904 non-null  int64
 13  Online boarding                   103904 non-null  int64
 14  Seat comfort                      103904 non-null  int64
 15  Inflight entertainment            103904 non-null  int64
 16  On-board service                  103904 non-null  int64
 17  Leg room service                  103904 non-null  int64
 18  Baggage handling                  103904 non-null  int64
 19  Checkin service                   103904 non-null  int64
 20  Inflight service                  103904 non-null  int64
 21  Cleanliness                       103904 non-null  int64
 22  Departure Delay in Minutes        103904 non-null  int64
 23  Arrival Delay in Minutes          103594 non-null  float64
 24  satisfaction                      103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

d. *.isna().sum()*

Dalam melakukan pembersihan data, harus dilakukan pengecekan data yang memiliki nilai *null* dan harus dilakukan *delete* terhadap baris data tersebut. Pada penelitian ini digunakan fungsi *isna()* yang mengembalikan nilai *boolean*

untuk mengetahui kolom yang memiliki nilai *null* dan *sum()* untuk menghitung total nilai null yang terdapat dalam *dataframe*.

e. *.dropna*

Berdasarkan perhitungan terdapat 310 baris dengan nilai *null*, maka peneliti membuang baris tersebut menggunakan fungsi *dropna*.

```
In [6]: # men-drop missing values yang ada pada dataframe.
        df_eda.dropna(inplace=True)
```

2.2 *Exploratory Data Analysis*

Exploratory Data Analysis digunakan oleh peneliti untuk menggali lebih dalam mengenai *dataset* yang digunakan. Peneliti mengelompokkan kolom berdasarkan tipe kolom yaitu *categorical, numerical*, dan *ordinal*.

2.3 *Modelling*

Pada bagian ini merupakan bagian prediksi penelitian untuk menentukan kepuasan pelanggan pada saat menggunakan maskapai penerbangan. Modelling menggunakan berbagai metode antara lain:

a. *Feature Engineering*

Pada tahap *feature engineering* dilakukan penghapusan kolom yang dianggap tidak berkorelasi yaitu kolom nama. Selain itu, dilakukan juga *label encoding* yang dapat mengubah *data categorical* menjadi data numerik untuk memudahkan pemodelan. Peneliti menggunakan fungsi *LabelEncoder()* pada lima kolom yaitu kolom *gender, customer_type, typer_of_travel, class,* dan *satisfaction*. Setelah itu, *dataframe* dipisahkan menjadi 2 yaitu X dan y, dimana X menyimpan kolom variabel bebas dan y menyimpan variabel target.

b. *Fitting*

Pada tahap *fitting*, peneliti menggunakan fungsi *train_test_split* yang terdapat pada *library sklearn.model_selection*. Dengan menggunakan fungsi tersebut, peneliti dapat membagi *dataset* menjadi 4 variabel yaitu *X_train,*

X_test, y_train, y_test, dengan ukuran variabel *test* 20% dari dataset utama serta *random_state* yang digunakan adalah 128.

c. *Light GBM*

Light gradient boosting machine yang disingkat *light GBM* adalah sebuah algoritma *machine learning* dengan teknik *boosting* untuk menghasilkan model prediksi yang lebih akurat. *Boosting* adalah proses dimana algoritma menggabungkan beberapa model yang lebih sederhana menjadi satu model yang lebih kuat. Keunggulan dari algoritma *Light GBM* adalah kecepatan dan efektivitasnya, sehingga dapat menangani data dalam jumlah yang besar, fitur yang banyak, dan terdistribusi secara tidak merata.

BAB III
HASIL DAN DISKUSI

3.1 Hasil awal

Hasil dibawah ini, merupakan hasil menggunakan LightGBM yang sudah di scaling dan tidak di scaling. Hasil dari keduanya memiliki akurasi yang sama yaitu 96%. Oleh karena itu, kami memutuskan untuk memilih yang telah di scaling karena jika data tidak di scaling maka data tersebut akan memiliki skala yang beragam. Dengan adanya scaling, data memiliki skala yang sama dengan begitu algoritma machine learning akan berjalan efektif, efisien, dan dan tidak mengganggu perhitungan yang dilakukan oleh algoritma.

LightGBM (Scale)

```
mislabel : 707
96.41967721724455
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     12019
           1       0.94      0.98      0.96      8700

    accuracy                           0.97     20719
   macro avg       0.96      0.97      0.97     20719
weighted avg       0.97      0.97      0.97     20719

[[11525   494]
 [  213  8487]]
```

LightGBM(Not Scale)

```
mislabel : 707
96.41967721724455
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     12019
           1       0.94      0.98      0.96      8700

    accuracy                           0.97     20719
   macro avg       0.96      0.97      0.97     20719
weighted avg       0.97      0.97      0.97     20719

[[11525   494]
 [  213  8487]]
```

3.2 Fitting Train Dataset

Kami menggunakan 4 metode machine learning (CatBoost, Random Forest, Gradient Boosting, LightGBM) untuk memberikan hasil dari dataset train. Berdasarkan hasil yang telah diberikan, metode LightGBM dan GradientBoosting memberikan akurasi maksimal sebesar 97% dibandingkan dengan CatBoost dan RandomForest yang hanya 96%.

LightGBM(Scaled, Train)

```
mislabel : 707
96.41967721724455
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     12019
           1       0.94      0.98      0.96      8700

    accuracy                           0.97     20719
   macro avg       0.96      0.97      0.97     20719
weighted avg       0.97      0.97      0.97     20719

[[11525   494]
 [  213  8487]]
```

CatBoost(Scaled, Train)

```
mislabel : 740
cross val score : 95.9486989058252
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     11952
           1       0.95      0.97      0.96      8767

    accuracy                           0.96     20719
   macro avg       0.96      0.97      0.96     20719
weighted avg       0.96      0.96      0.96     20719

[[11475   477]
 [  263  8504]]
```

Random Forest (Scaled, Train)

```
mislabel : 738
cross val score : 95.9486989058252
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     12018
           1       0.94      0.97      0.96      8701

    accuracy                           0.96     20719
   macro avg       0.96      0.97      0.96     20719
weighted avg       0.96      0.96      0.96     20719

[[11509   509]
 [  229  8472]]
```

Gradient Boosting (Scale, Train)

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     11738
           1       0.97      0.95      0.96      8981

    accuracy                           0.97     20719
   macro avg       0.97      0.96      0.96     20719
weighted avg       0.97      0.97      0.97     20719
```

3.3 Fitting Test Dataset

Berdasarkan metode yang kita pilih yaitu LightGBM dan Gradient Boosting dengan akurasi 97% di train data. Maka, kita melakukan evaluasi model pada data testing menggunakan dua metode tersebut. Evaluasi dari classification report menunjukkan bahwa kedua metode memiliki akurasi akhir 96% dengan data testing.

Light GBM (Scale, Test)

```
mislabel : 930
cross val score : 95.9486989058252
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     14528
           1       0.97      0.94      0.96     11365

    accuracy                           0.96     25893
   macro avg       0.97      0.96      0.96     25893
weighted avg       0.96      0.96      0.96     25893

[[14246   648]
 [  282 10717]]
```

Gradient Boosting (Scale, Test)

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     14528
           1       0.97      0.94      0.96     11365

    accuracy                           0.96     25893
   macro avg       0.96      0.96      0.96     25893
weighted avg       0.96      0.96      0.96     25893

[[14206   322]
 [  642 10723]]
```

3.4 Hasil Akhir

Kami memutuskan untuk menggunakan metode Light GBM karena confusion metricsnya memiliki nilai true positif yang lebih banyak daripada Gradient Boosting. Dengan menggunakan model Light GBM, kami memprediksikan hasil sebanyak 14246 pelanggan netral/tidak puas dengan maskapai penerbangan dan sebanyak 10717 pelanggan puas dengan maskapai penerbangan yang digunakan. Presisi dari prediksi ini sebesar 96% pada pelanggan netral/tidak puas dan sebesar 97% pada pelanggan puas.

F1-Score yang didapat keduanya yaitu sebesar 97% dan 96% sehingga menunjukkan bahwa model prediksi yang dibuat merupakan model yang akurat dalam memprediksikan kepuasan pelanggan. Akurasi akhir dari model LightGBM adalah 96%.

```
mislabel : 930
cross val score : 95.9486989058252
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     14528
           1       0.97      0.94      0.96     11365

    accuracy                           0.96     25893
   macro avg       0.97      0.96      0.96     25893
weighted avg       0.96      0.96      0.96     25893

[[14246   648]
 [  282 10717]]
```

Gambar 1. *Classification report* pada prediksi akhir

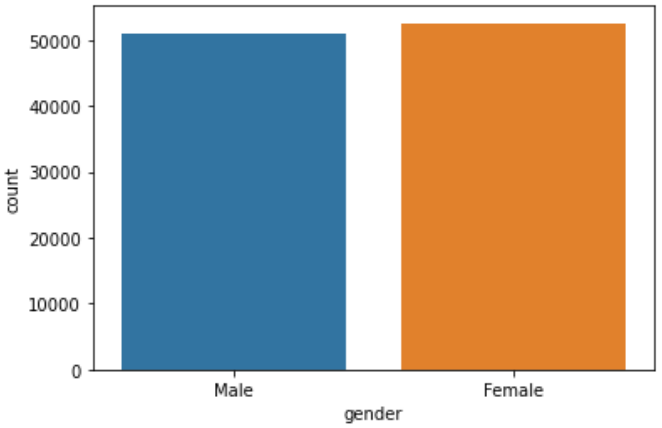
BAB IV
KESIMPULAN

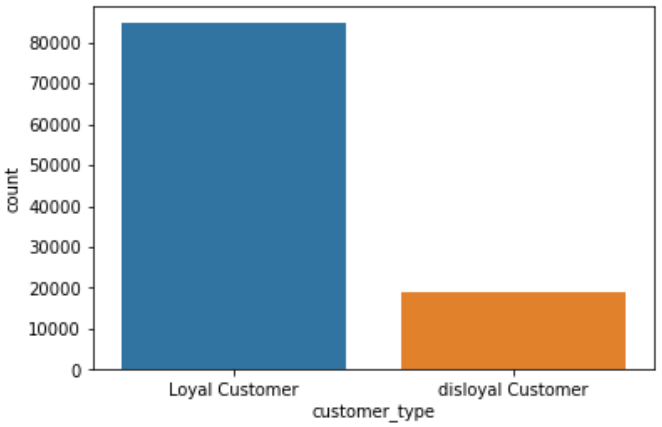
Setelah melalui proses analisis yang metodologis dan sistematis, yang meliputi metode evaluasi performa *classification_report()*, dapat dikonklusikan bahwa algoritma LightGBM merupakan solusi yang paling optimal dibandingkan dengan metode lain seperti XGBoost, CatBoost, dan RandomForest dalam menyelesaikan masalah yang dihadapi dalam proyek ini. Evaluasi performa yang dilakukan menyediakan bukti yang menunjukkan bahwa algoritma LightGBM memiliki tingkat akurasi yang signifikan, yaitu sebesar 96% pada data uji yang digunakan.

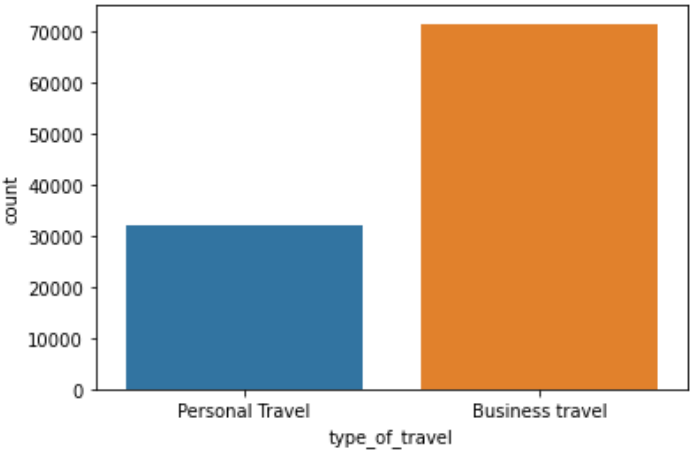
LAMPIRAN

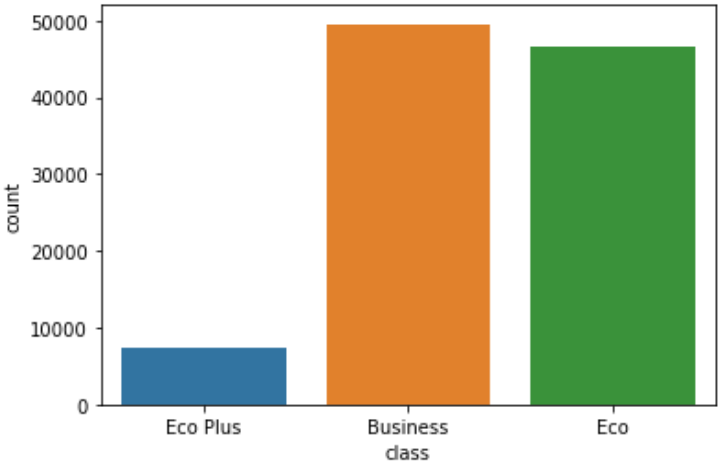
Lampiran 1 : Exploratory Data Analysis

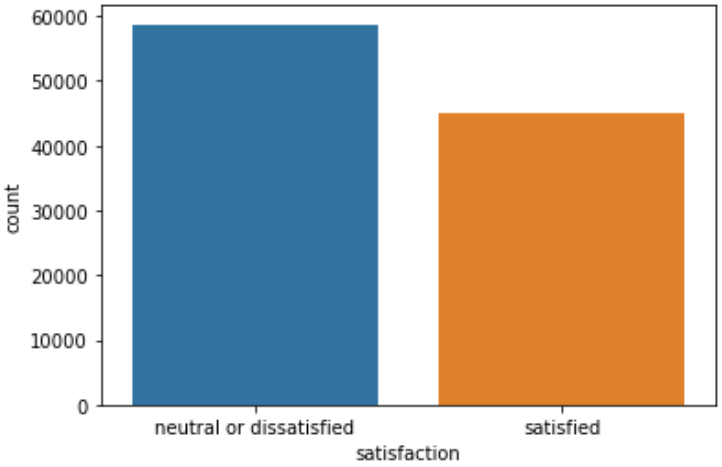
```
In [14]: for x in categorical:
             ax = sns.countplot(x, data=df_eda)
             plt.show()
```

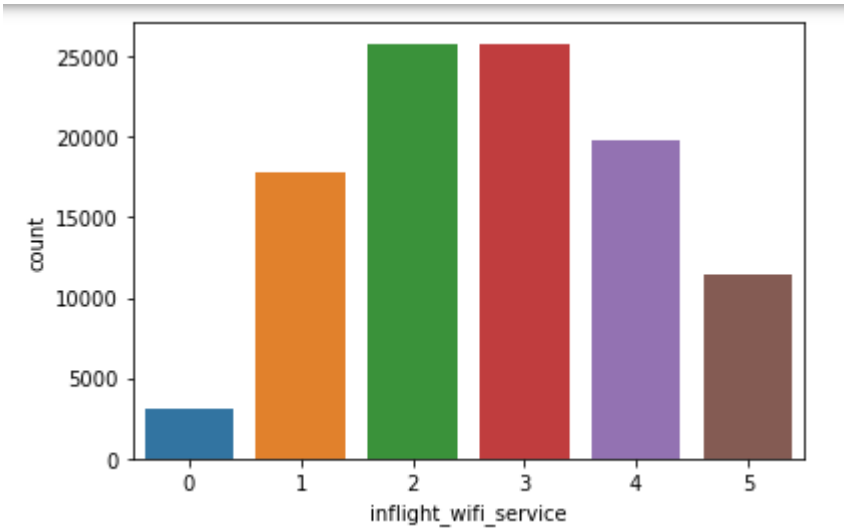


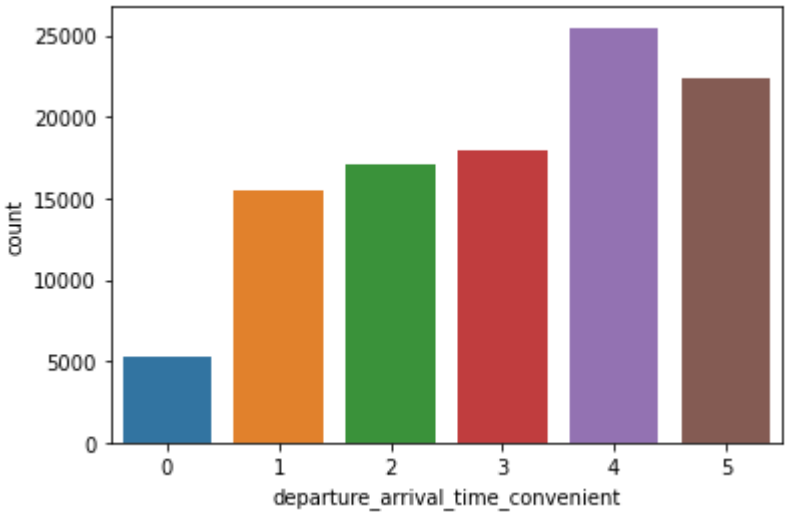


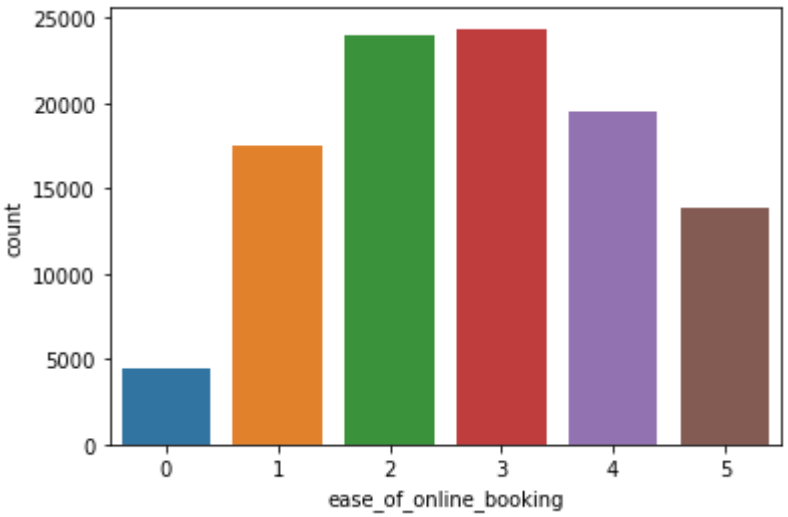


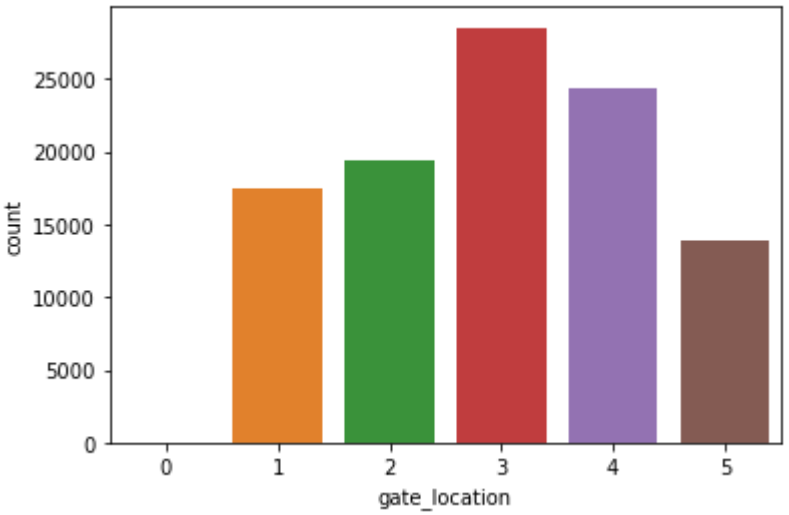

50000
40000
30000
20000
10000
0
count
Eco Plus
Business
Eco
class

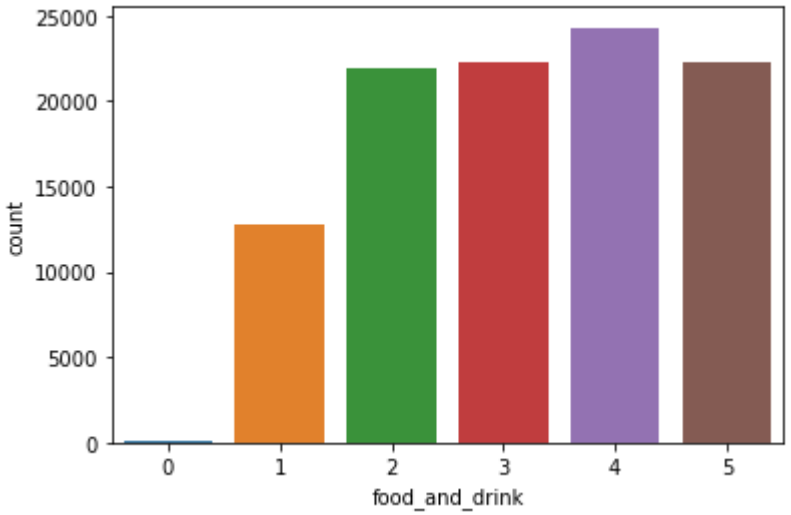

60000
50000
40000
30000
20000
10000
0
count
neutral or dissatisfied
satisfied
satisfaction

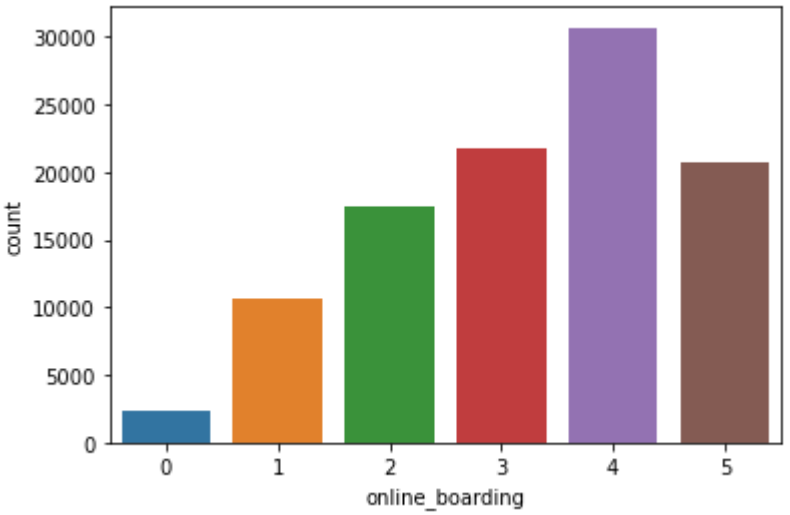

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
inflight_wifi_service

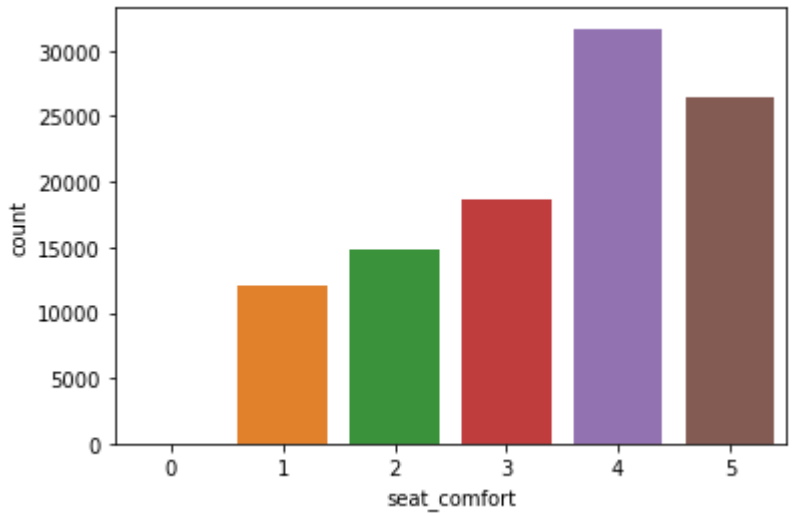

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
departure_arrival_time_convenient

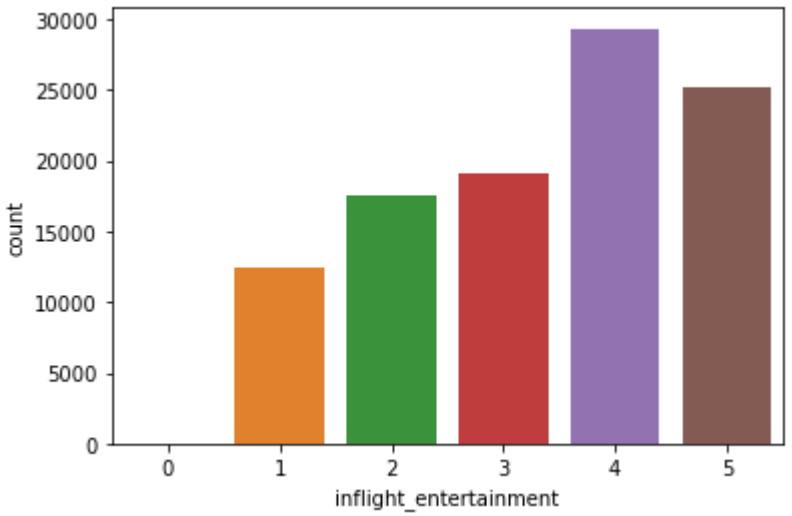

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
ease_of_online_booking

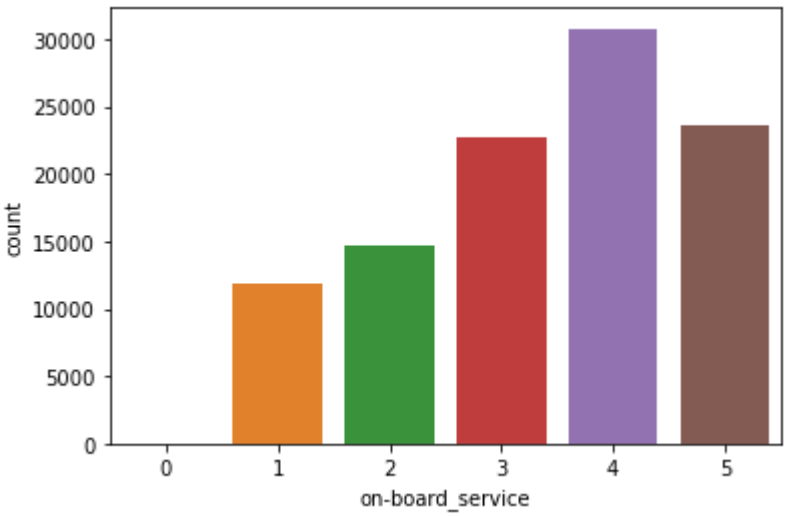

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
gate_location

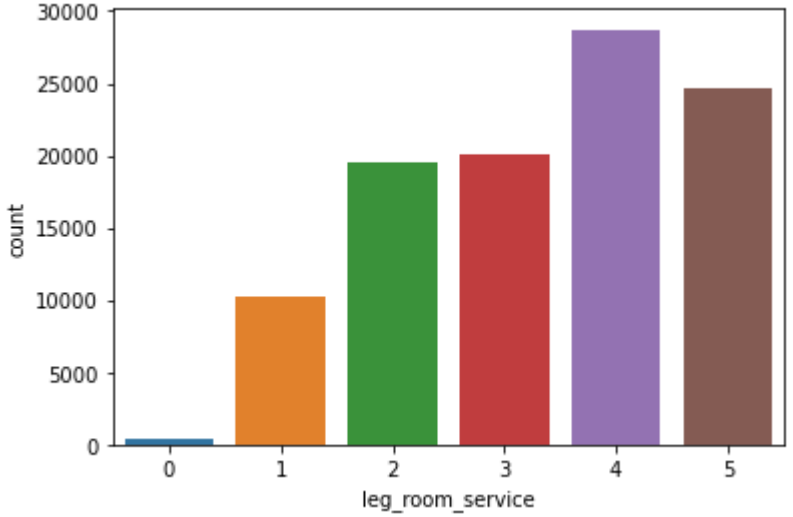

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
food_and_drink

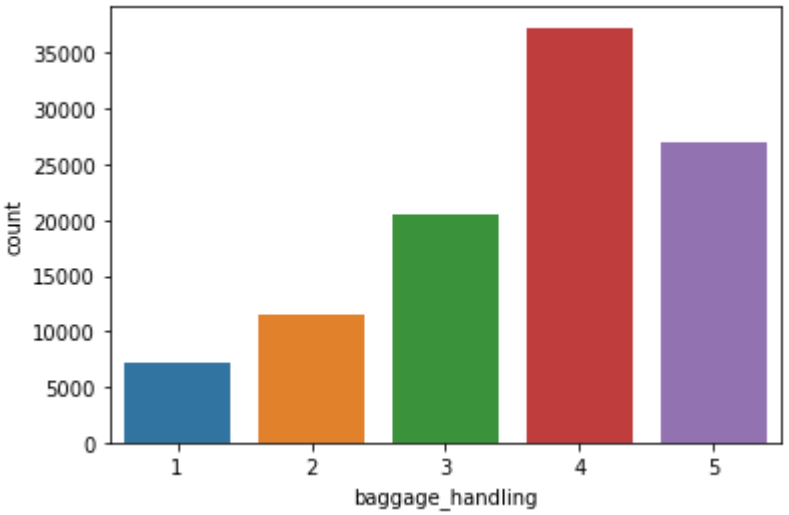

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
online_boarding

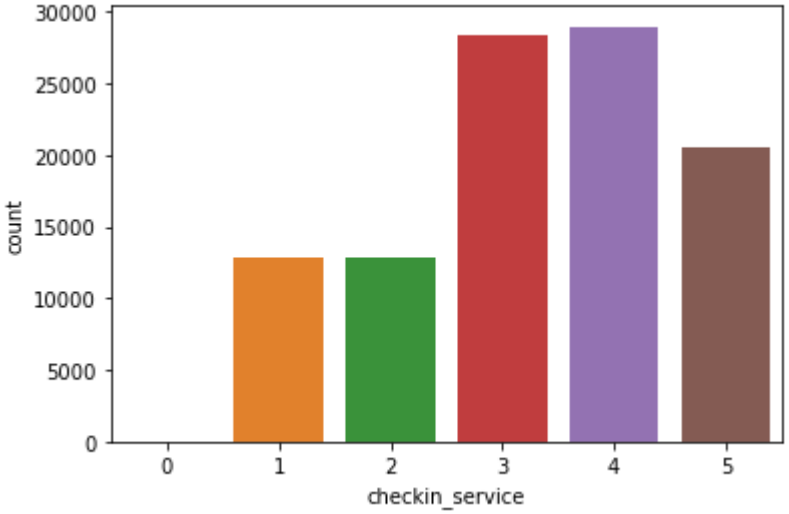

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
seat_comfort

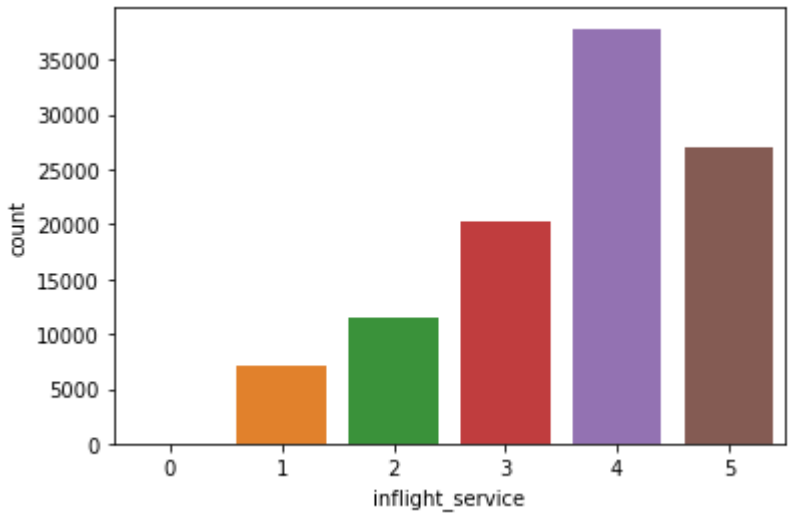

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
inflight_entertainment

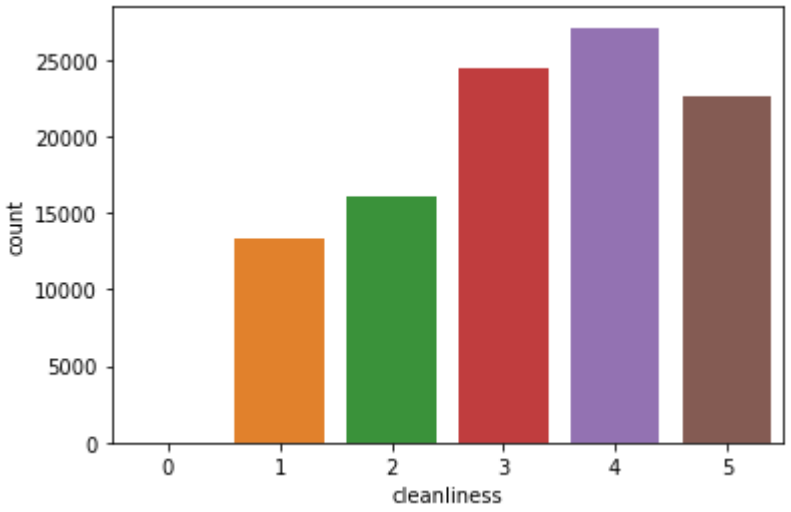

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
on-board_service

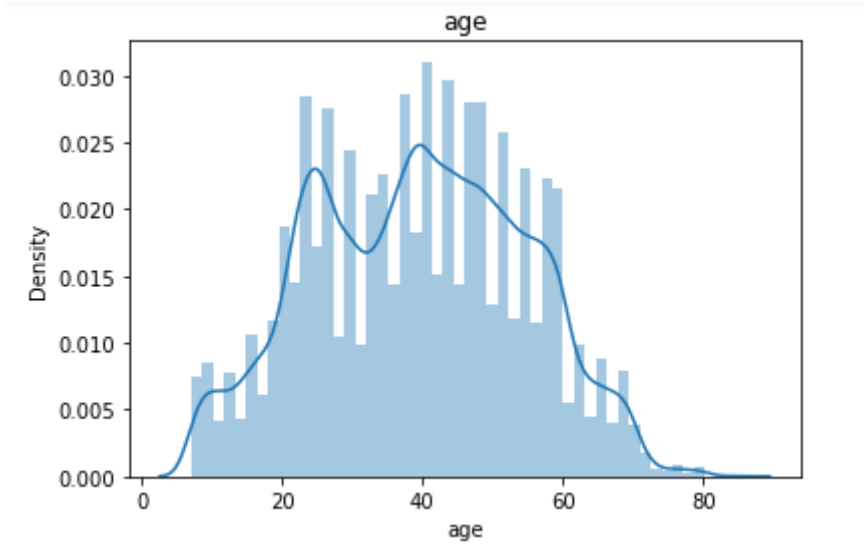

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
leg_room_service

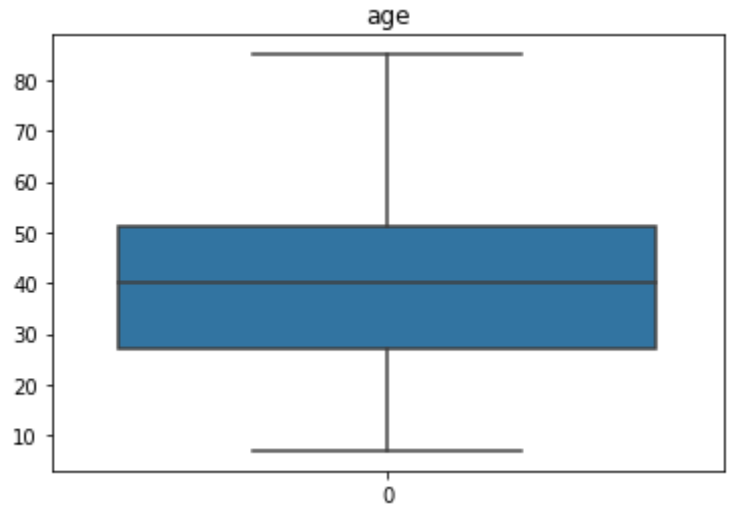

35000
30000
25000
20000
15000
10000
5000
0
count
1
2
3
4
5
baggage_handling

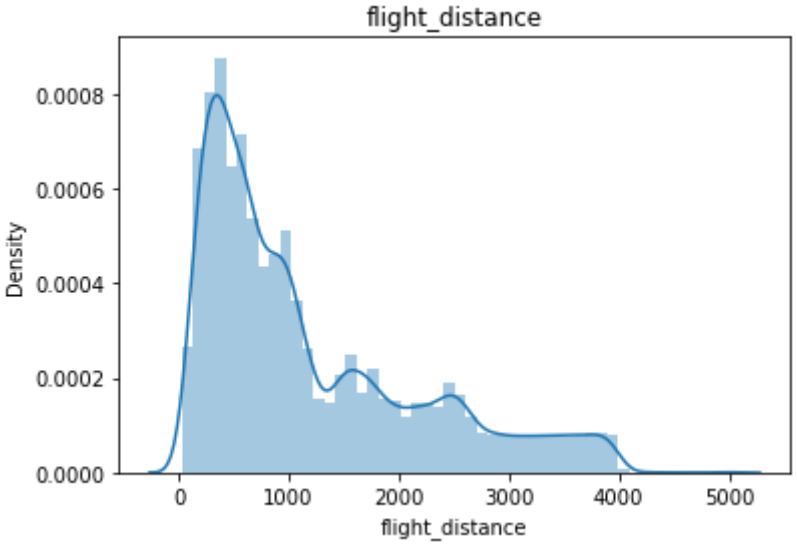

30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
checkin_service

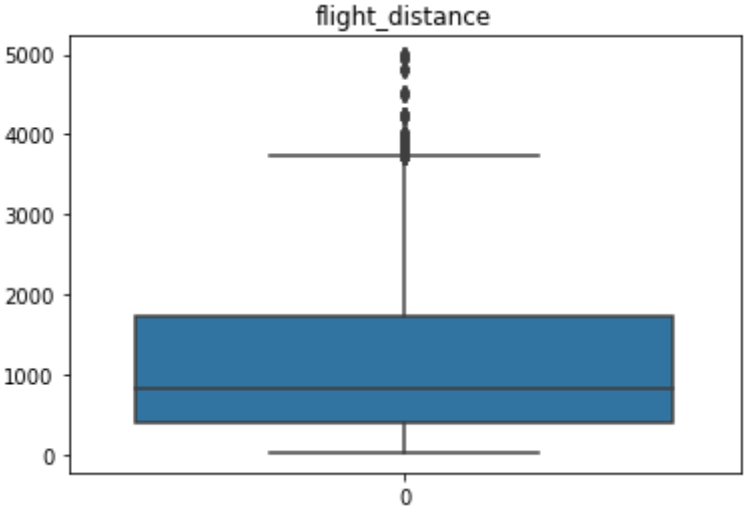

35000
30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
inflight_service

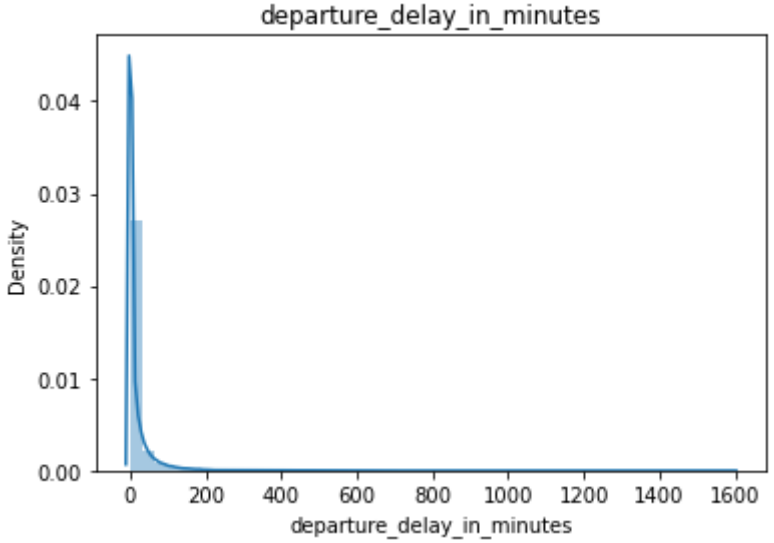

25000
20000
15000
10000
5000
0
count
0
1
2
3
4
5
cleanliness

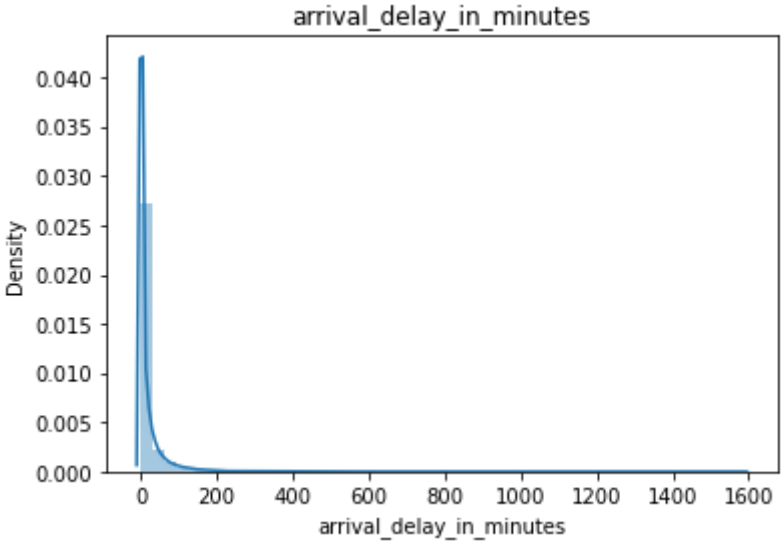

age
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
20
40
60
80
age


age
80
70
60
50
40
30
20
10
0


flight_distance
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
1000
2000
3000
4000
5000
flight_distance


flight_distance
5000
4000
3000
2000
1000
0
0


departure_delay_in_minutes
Density
0.04
0.03
0.02
0.01
0.00
0
200
400
600
800
1000
1200
1400
1600
departure_delay_in_minutes


arrival_delay_in_minutes
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
200
400
600
800
1000
1200
1400
1600
arrival_delay_in_minutes