## Opis danych

Zbiór danych pochodzi z ankiet przeprowadzanych pasażerom lotów – badano ich poziom satysfakcji z lotu. W zbiorze danych znajdują się wszelkie niezbędne informacje dotyczące pasażera, samolotu a także obsługi podczas lotu.

**Teza: Czy pasażer był zadowolony z lotu?**

**Możliwe wyniki: Neutral or Dissatisfied (0), Satisfied(1)**

Opis danych znajdujących się w zbiorze (postać zbioru przed EDA):

**Gender (Płeć):** Płeć pasażera (Female, Male)

**Customer Type (Typ pasażera):** Typ pasażera (Loyal customer, disloyal customer)

**Age (Wiek):** Wiek pasażera

**Type of Travel (Powód podróży):** Powód podróży (Personal Travel, Business Travel)

**Class (Klasa):** Klasa lotu (Business, Eco, Eco Plus)

**Flight distance (Dystans lotu):** Dystans lotu w kilometrach

**Inflight wifi service**: Poziom satysfakcji z jakości sieci Wi-Fi (0:Nie dotyczy;1-5)

**Departure/Arrival time convenient:** Poziom satysfakcji z czasu przylotu/odlotu (0:Nie dotyczy;1-5)

**Ease of Online booking:** Poziom satysfakcji z obsługi rezerwacji biletu online (0:Nie dotyczy;1-5)

**Gate location:** Poziom satysfakcji z położenia miejsca odlotu (0:Nie dotyczy;1-5)

**Food and drink:** Poziom satysfakcji z jedzenia i napoi dostępnych na pokładzie (0:Nie dotyczy;1-5)

**Online boarding:** Poziom satysfakcji z obsługi zamówienia lotu online (0:Nie dotyczy;1-5)

**Seat comfort:** Poziom satysfakcji z komfortu siedziska (0:Nie dotyczy;1-5)

**Inflight entertainment:** Poziom satysfakcji z rozrywki dostępnej podczas lotu (0:Nie dotyczy;1-5)

**On-board service:** Poziom satysfakcji z obsługi pasażera podczas odlotu (0:Nie dotyczy;1-5)

**Leg room service:** Poziom satysfakcji z ilości miejsca na nogi (0:Nie dotyczy;1-5)

**Baggage handling:** Poziom satysfakcji z obsługi bagażu (0:Nie dotyczy;1-5)

**Check-in service:** Poziom satysfakcji z obsługi przylotu (0:Nie dotyczy;1-5)

**Inflight service:** Poziom satysfakcji z obsługi pasażera podczas lotu (0:Nie dotyczy;1-5)

**Cleanliness:** Poziom satysfakcji z czystości w samolocie (0:Nie dotyczy;1-5)

**Departure Delay in Minutes:** O ile opóźnił się odlot (w minutach)

**Arrival Delay in Minutes:** O ile opóźnił się przylot (w minutach)

**Satisfaction:** Target – czy pasażer był zadowolony (Satisfaction, neutral or dissatisfaction

## EDA

1. Stan danych przed EDA:

Shape: (129880, 25)

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)

**Lista brakujących wartości w kolumnach:**

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 310

satisfaction 0

1. **Pozbycie się niepotrzebnych (nic nie wnoszących) kolumn:**

Pozbywam się pierwszej i drugiej kolumny - „Unnamed” oraz „id”, jako że nie mają żadnego wpływu na wynik.

df = df.drop(columns=[df.columns[0], "id"], axis=1)

1. **Pozbycie się komórek, które nie posiadają żadnej wartości:**

Jedyną kolumną, która posiadała puste komórki była „Arrival Delay in Minutes”. Zastępuję puste wartości zerami.

df.fillna(0, inplace=True)

1. **Zmiana typu danych z tekstu (typ object) na wartości liczbowe**

**W tym celu skorzystam z LabelEncodera z biblioteki sklearn. Najpierw sporządzam tablicę z nazwami kolumn, których wartości chcę zmienić:**

attr\_object\_list = []  
for col in df.columns:  
 if df[col].dtype=="object":  
 attr\_object\_list.append(col)

**Docelowe kolumny: ['Gender', 'Customer Type', 'Type of Travel', 'Class', 'satisfaction']**

**Tworzę funkcję przyjmującą jako argument dataset oraz nazwe kolumny, której wartości chcę zmienić na liczbowe:**

def Encode(df, col):  
 encoder = LabelEncoder()  
 encoder.fit(df[col])  
 new\_column = encoder.transform(df[col])  
  
 test = np.unique(new\_column)  
 result = encoder.inverse\_transform(test)  
 for i in range(len(test)):  
 print(result[i],":".format(),test[i])  
  
 return new\_column

**Stosuję stworzoną funkcję do każdej z kolumn, która wymaga zmiany wartości:**

for col in attr\_object\_list:  
 if df[col].dtype == "object":  
 df[col] = Encode(df, col)  
 else:

break

**Nowe wartości w w.w. kolumnach:**

for column in attr\_object\_list:  
 print ("\"", column, "\" values: ", df[column].unique())

*Female : 0*

*Male : 1*

*Loyal Customer : 0*

*disloyal Customer : 1*

*Business travel : 0*

*Personal Travel : 1*

*Business : 0*

*Eco : 1*

*Eco Plus : 2*

*neutral or dissatisfied : 0*

*satisfied : 1*

1. **Zamieniam typ danych w kolumnie „Arrival Delay in Minutes” z float na int:**

df["Arrival Delay in Minutes"] = df["Arrival Delay in Minutes"].astype(int)

1. **Pozbywam się kolumny „Departure Delay in Minutes” jako że zaobserwowałem silne powiązanie między nią, a kolumną „Arrival Delay in Minutes” na macierzy korelacji (macierz korelacji znajduje się na końcu podpunktu „Wykresy”)**

df = df.drop('Departure Delay in Minutes', axis=1)

1. Stan danych po EDA:

Shape: (129880, 22)

Data columns (total 23 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Gender 129880 non-null int32

1 Customer Type 129880 non-null int32

2 Age 129880 non-null int64

3 Type of Travel 129880 non-null int32

4 Class 129880 non-null int32

5 Flight Distance 129880 non-null int64

6 Inflight wifi service 129880 non-null int64

7 Departure/Arrival time convenient 129880 non-null int64

8 Ease of Online booking 129880 non-null int64

9 Gate location 129880 non-null int64

10 Food and drink 129880 non-null int64

11 Online boarding 129880 non-null int64

12 Seat comfort 129880 non-null int64

13 Inflight entertainment 129880 non-null int64

14 On-board service 129880 non-null int64

15 Leg room service 129880 non-null int64

16 Baggage handling 129880 non-null int64

17 Checkin service 129880 non-null int64

18 Inflight service 129880 non-null int64

19 Cleanliness 129880 non-null int64

20 Departure Delay in Minutes 129880 non-null int64

21 Arrival Delay in Minutes 129880 non-null int32

22 satisfaction 129880 non-null int32

dtypes: int32(6), int64(17)

**Lista brakujących wartości w kolumnach:**

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

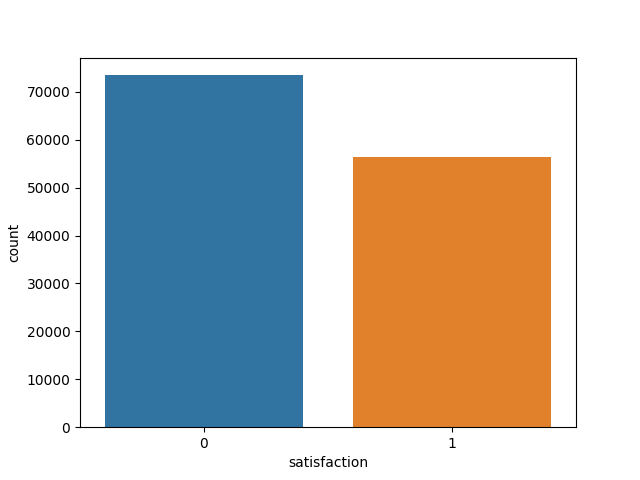
Departure Delay in Minutes 0

Arrival Delay in Minutes 0

satisfaction 0

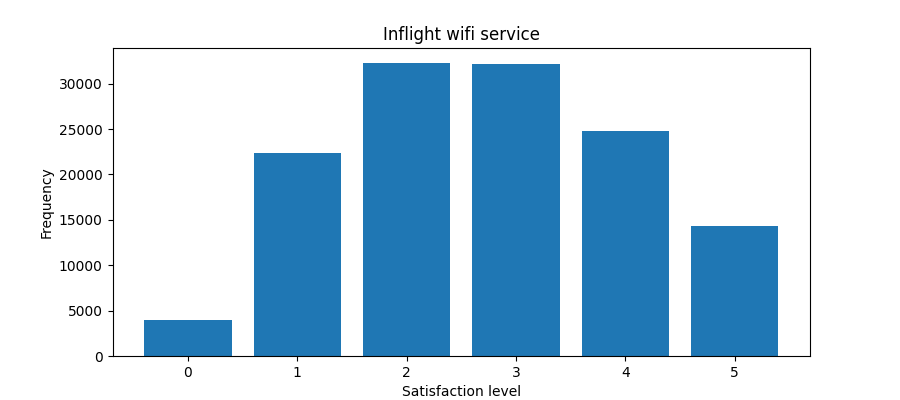
## Wykresy obrazujące dane:

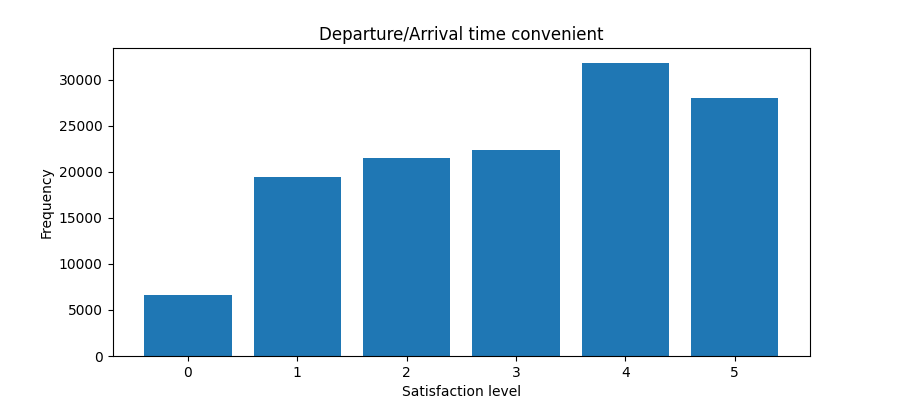
1. Wykres ukazujący poziom zadowolenia z lotu (Target):

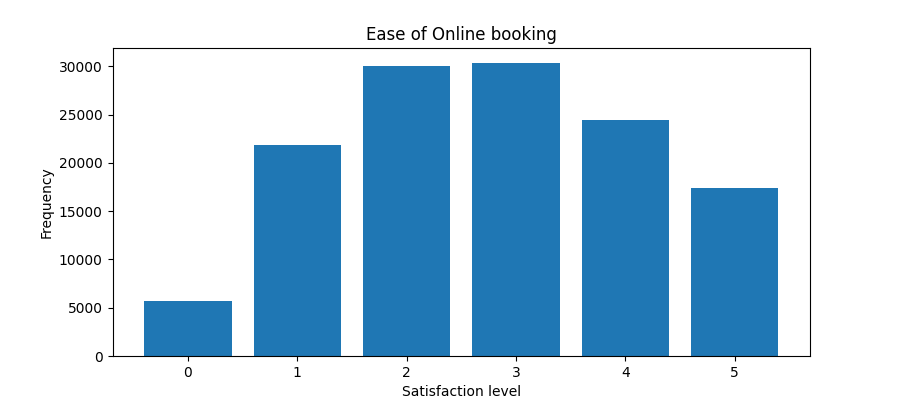


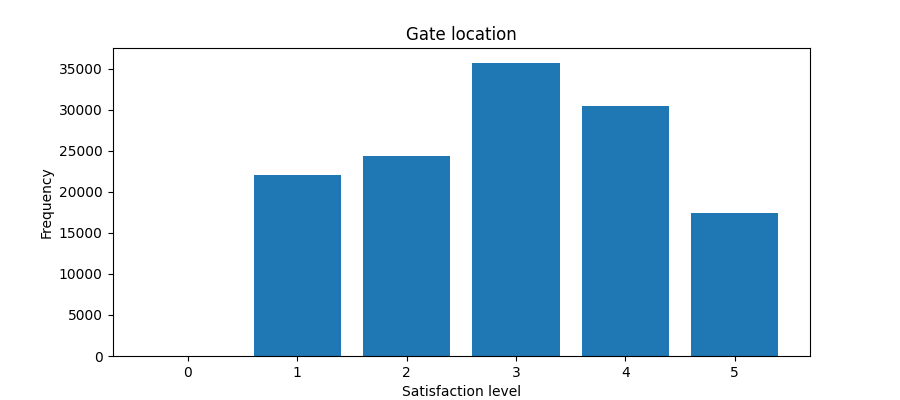
1. Wykresy obrazujące poziom satysfakcji z danej usługi (dane z ankiety)

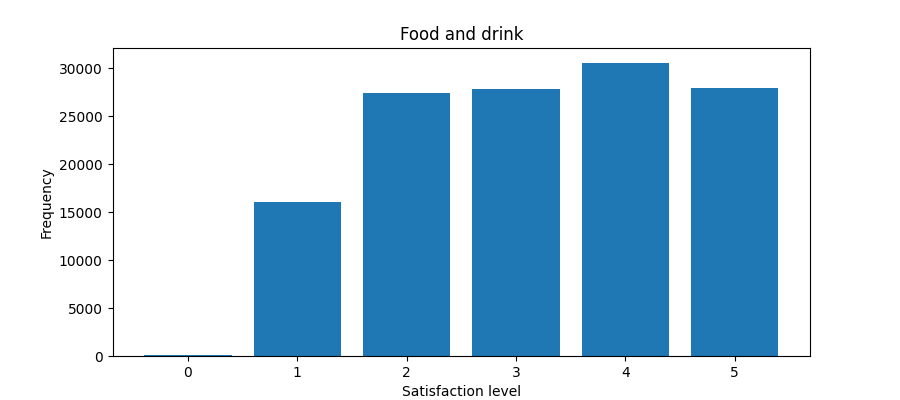
satisfaction\_columns = [ "Inflight wifi service", "Departure/Arrival time convenient", "Ease of Online booking", "Gate location", "Food and drink", "Online boarding", "Seat comfort", "Inflight entertainment", "On-board service", "Leg room service", "Baggage handling", "Checkin service", "Inflight service", "Cleanliness",]  
for col in satisfaction\_columns:  
 var = df[col]  
 var\_value = var.value\_counts()  
  
 plt.figure(figsize=(9, 3))  
 plt.bar(var\_value.index, var\_value.values)  
  
 plt.xlabel("Satisfaction level")  
 plt.ylabel("Frequency")  
 plt.title(col)  
 plt.show()

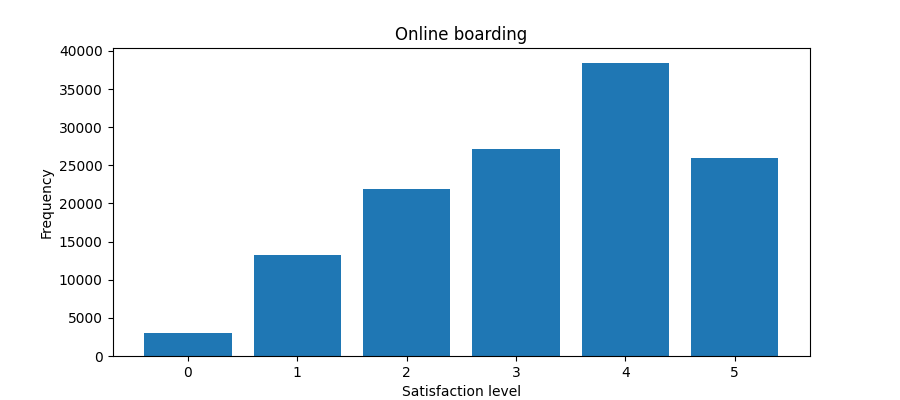


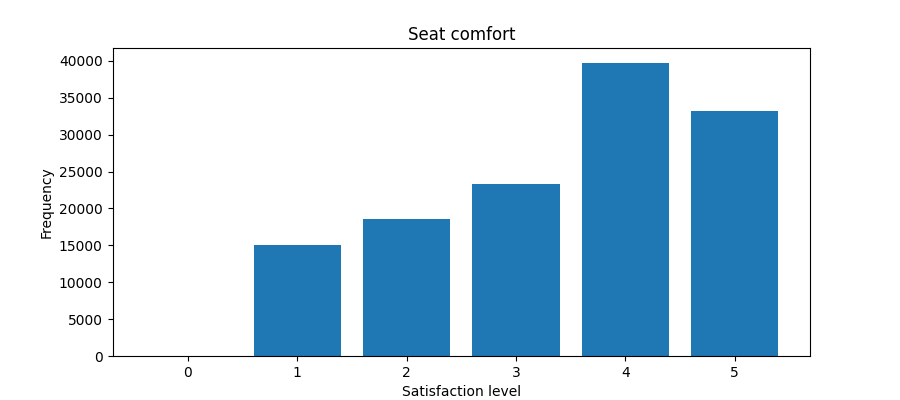


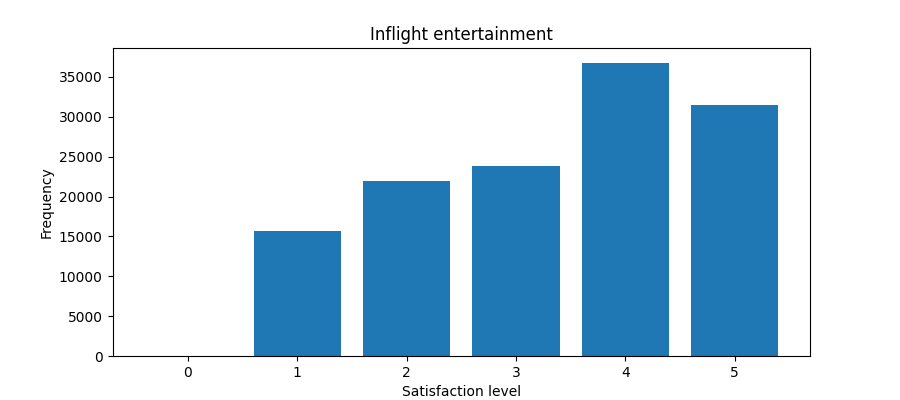


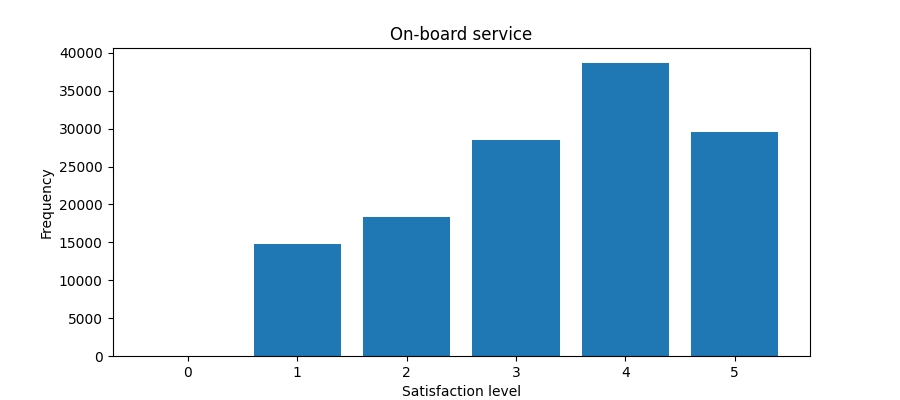


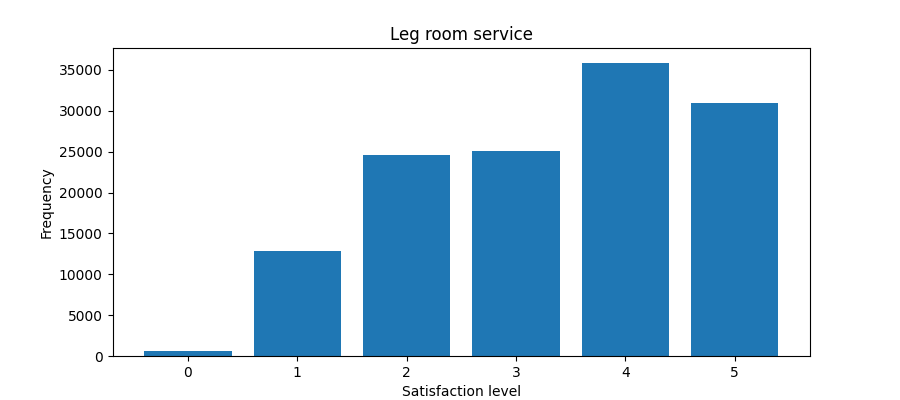


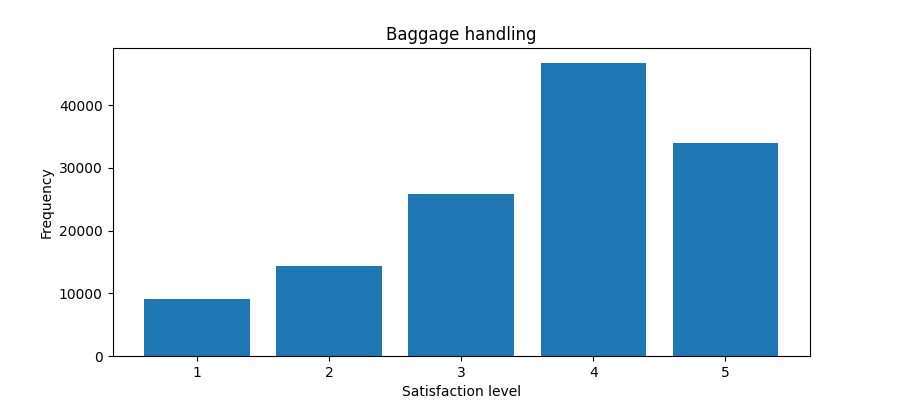


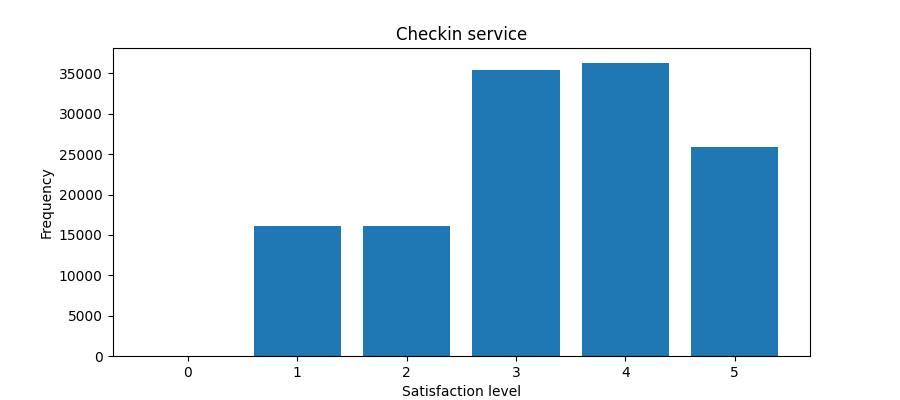


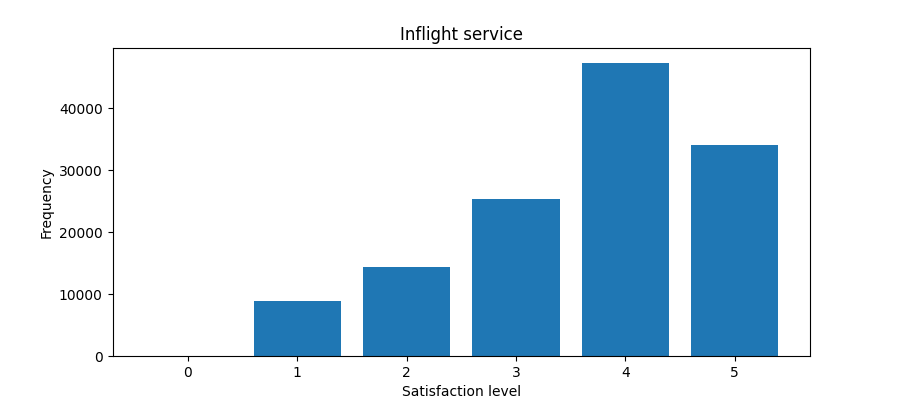




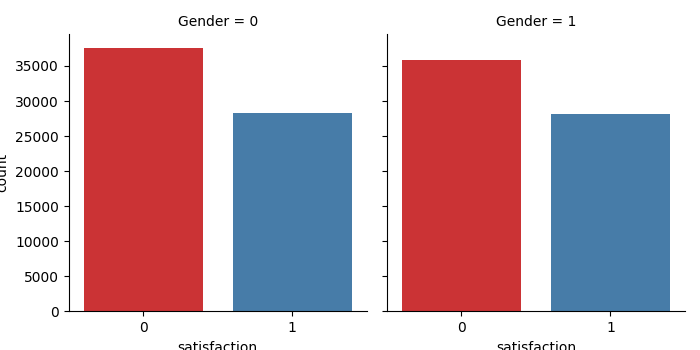




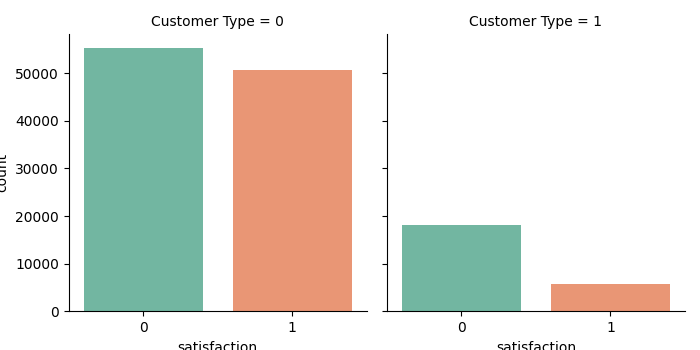




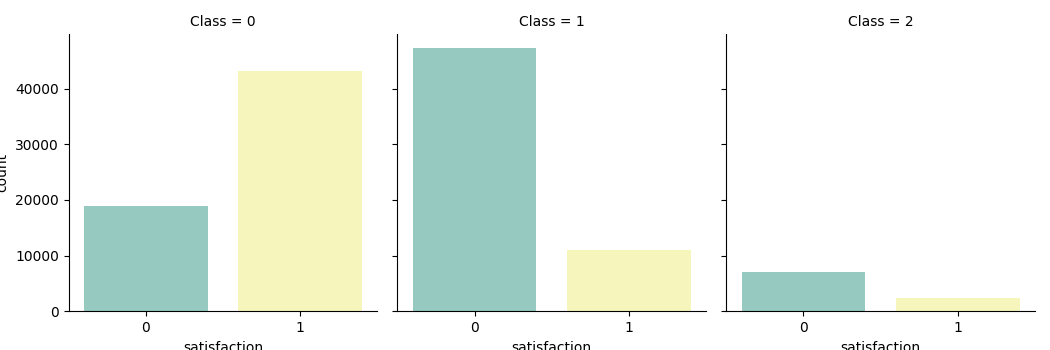
1. Wykres obrazujący poziom zadowolenia w zależności od płci (0 – mężczyzna, 1 – kobieta):



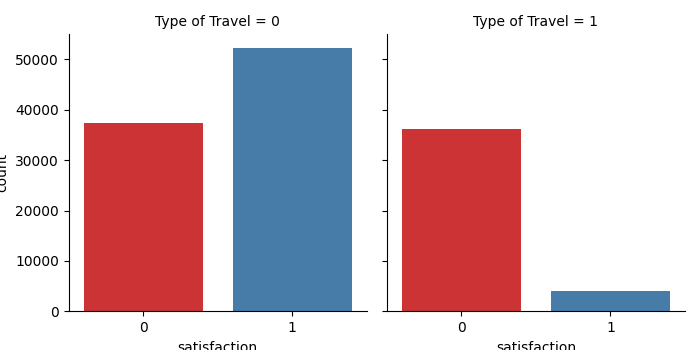
1. Wykres obrazujący poziom zadowolenia w zależności od typu pasażera (Typ pasażera, czyli stały klient (0), nowy klient (1))



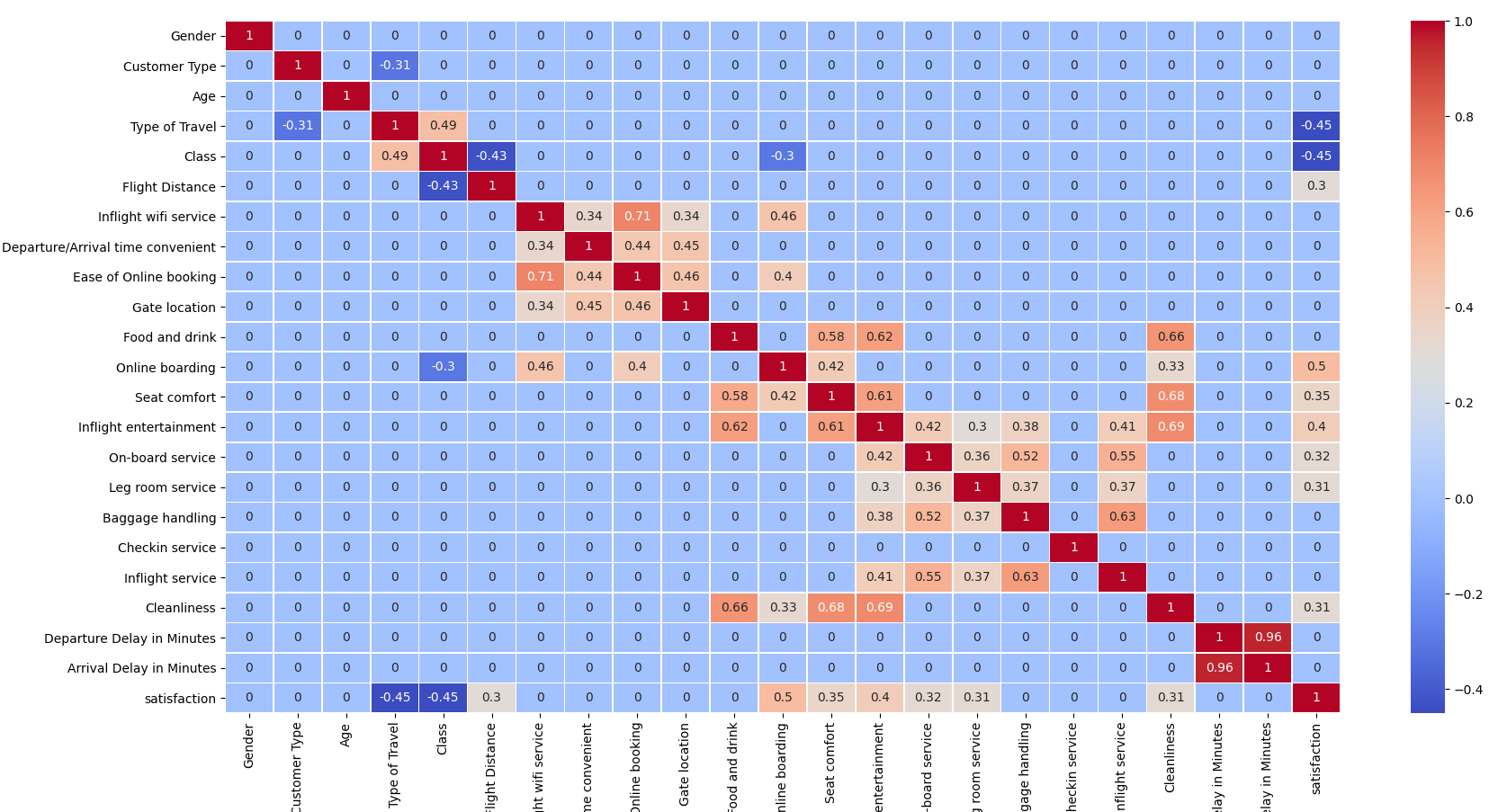
1. Wykres obrazujący poziom zadowolenia w zależności od klasy lotu (0 – Business, 1 – Eco, 2 – Eco Plus)



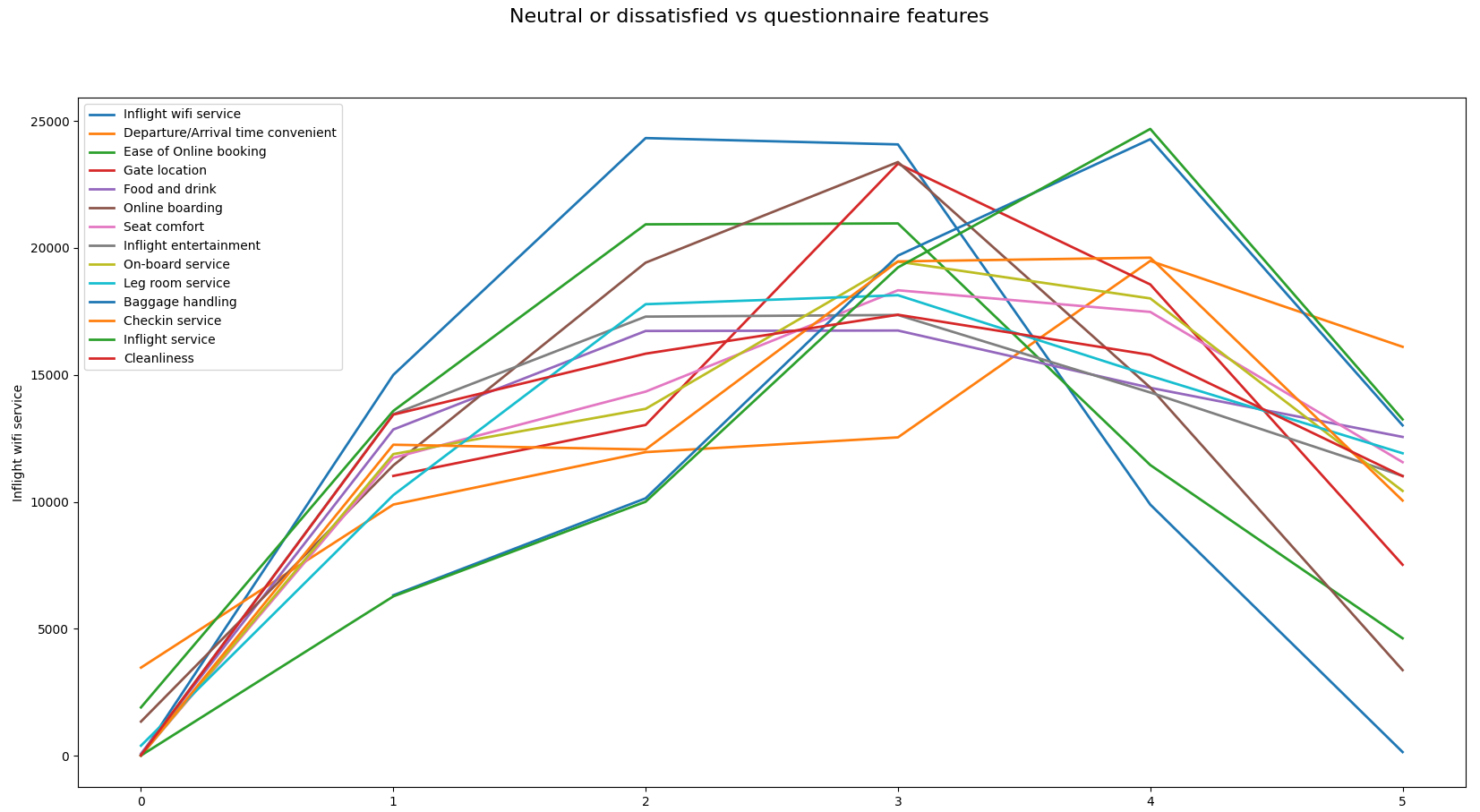
1. Wykres obrazujący poziom zadowolenia w zależności od typu lotu (Podróż służbowa – 0, Podróż niesłużbowa – 1)

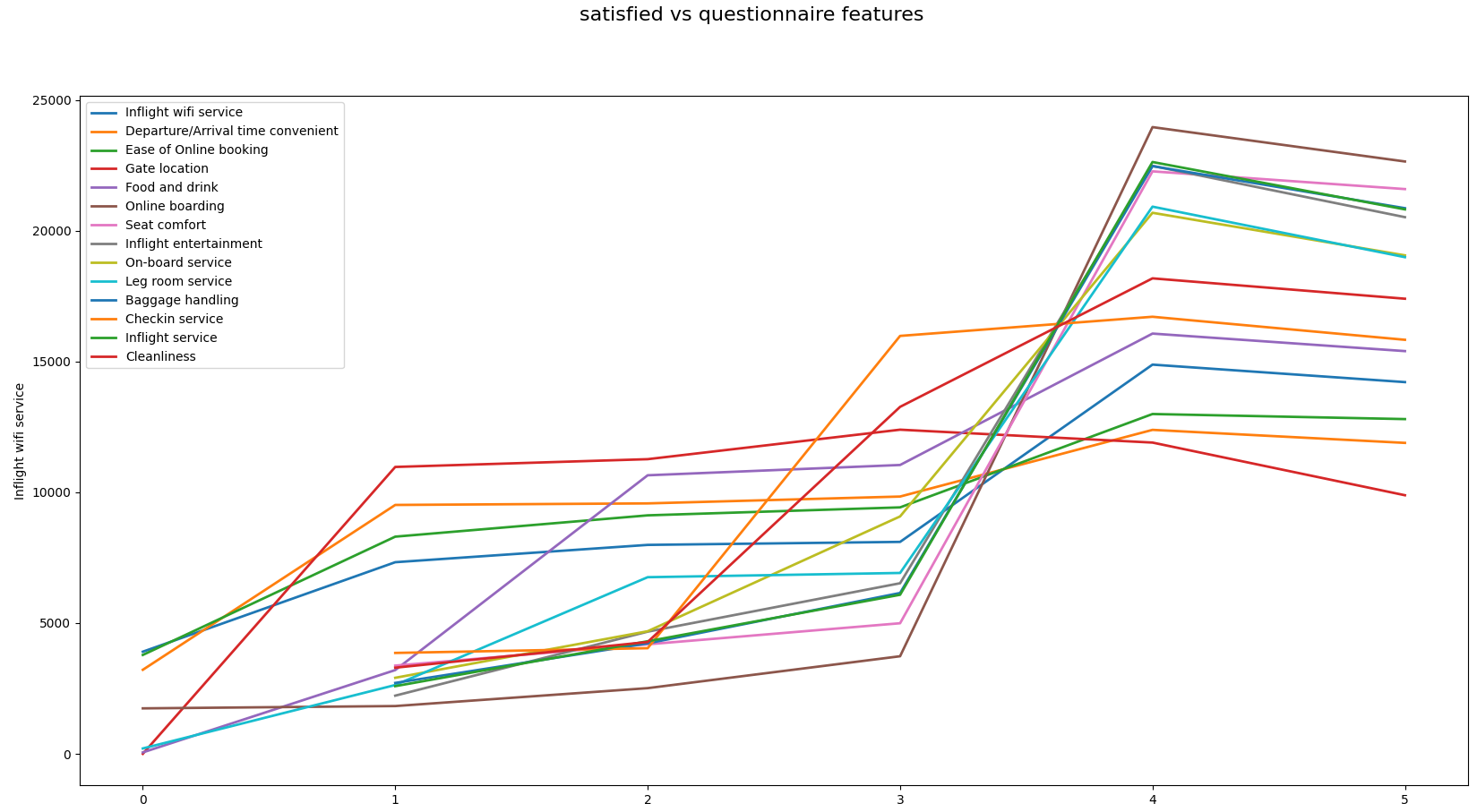


1. Macierz korelacji



Na macierzy korelacji możemy zaobserwować silne powiązanie pomiędzy czasami opóźnienia odlotu i przylotu (Kolumny „Departure Delay in Minutes” i „Arrival Delay in Minutes”). W związku z tym postanowiłem pozbyć się kolumny „Departure Delay in Minutes”).

1. Wykres liniowy ukazujący zależności między ocenionymi w ankiecie usługami do poziomu **niezadowolenia**
2. Wykres liniowy ukazujący zależności między ocenionymi w ankiecie usługami do poziomu **zadowolenia**



## Uczenie modeli i wyniki

Wybrałem 3 modele: LogisticRegression, RandomForestClassifier i MLPClassifier.

Do każdego z modeli przedstawiam cunfusion matrix porównujący przewidziane wartości do rzeczywistych.

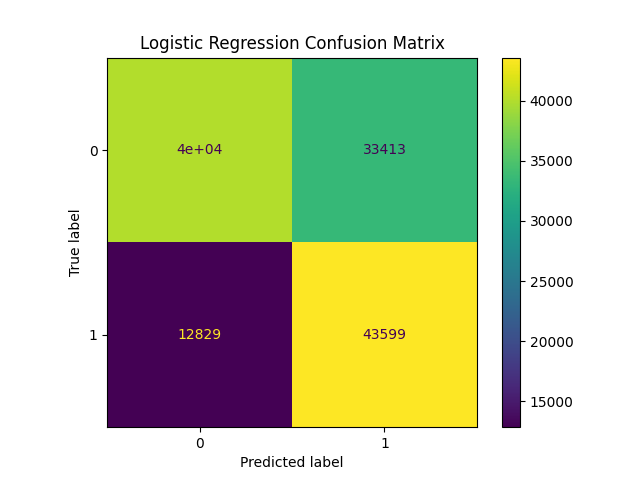
Aby znaleźć najlepsze parametry modeli, użyłem techniki CrossValidation.

Przed rozpoczęciem uczenia modeli, dokonałem normalizacji wartości w zbiorze danych:

scaler = StandardScaler()  
X\_train = scaler.fit\_transform(X\_train)  
X\_test = scaler.fit\_transform(X\_test)

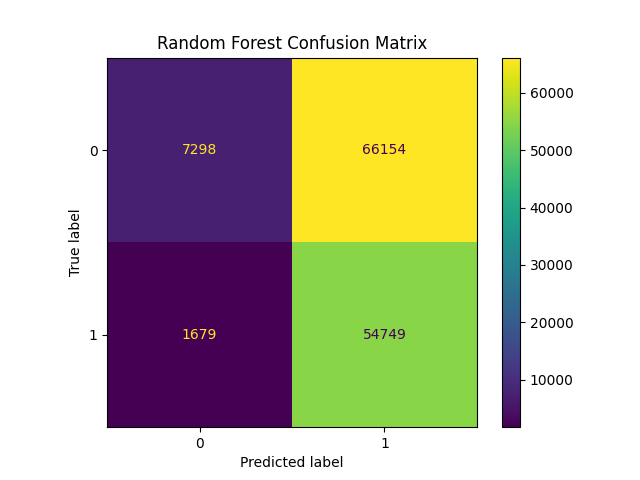
1. Logistic Regression

Logistic Regression Mean Accuracy (post cross-validation): **0.87441484447182**



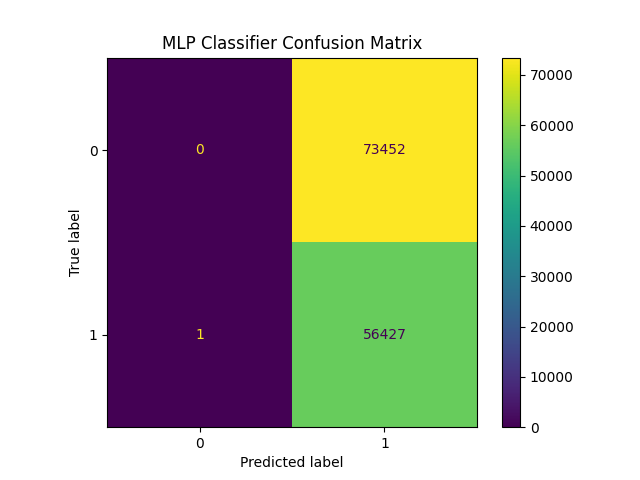
1. RandomForestClassifier

Random Forest Mean Accuracy (post cross-validation): **0.9636818601786266**

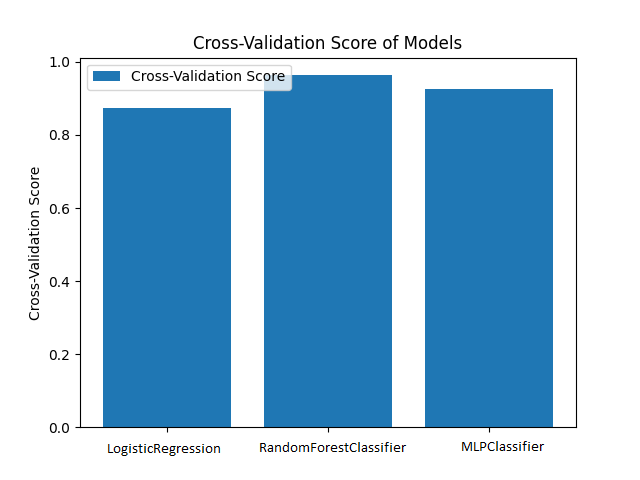
****

1. MLPClassifier

MLP Classifier Mean Accuracy (post cross-validation): **0.9247382198952879**



Wykres porównujący średnie wyniki uzyskane przez użyte modele:



Z pośród 3 użytych modeli, najlepiej poradził sobie RandomForestClassifier, jednak uczenie tego modelu trwało zdecydowanie najdłużej.