IUM Dokumentacja Końcowa

Rafał Budnik 318639
Ireneusz Okniński 310228

1. **Temat**

   "Jakiś czas temu wprowadziliśmy konta premium, które uwalniają użytkowników od słuchania reklam. Nie są one jednak jeszcze zbyt popularne – czy możemy się dowiedzieć, które osoby są bardziej skłonne do zakupu takiego konta?"

2. **Cele oraz założenia**

   a) **Cel biznesowy**

   Zwiększenie popularności kont premium poprzez identyfikację cech użytkowników, które są powiązane z większą skłonnością do zakupu tych kont, co przełoży się na większe zyski.

   b) **Zadanie modelowania**

   Ocena czy dany klient jest skłonny do zakupu pakietu premium. Skorzystamy z różnych modeli klasyfikacyjnych w celu znalezienia najbardziej skutecznego. Zadaniem modelowania będzie klasyfikacja binarna (będziemy oceniać, czy użytkownik nabędzie premium bądź nie).

   c) **Analityczne kryterium sukcesu**

   Poprawienie wyników klasyfikatora naiwnego:

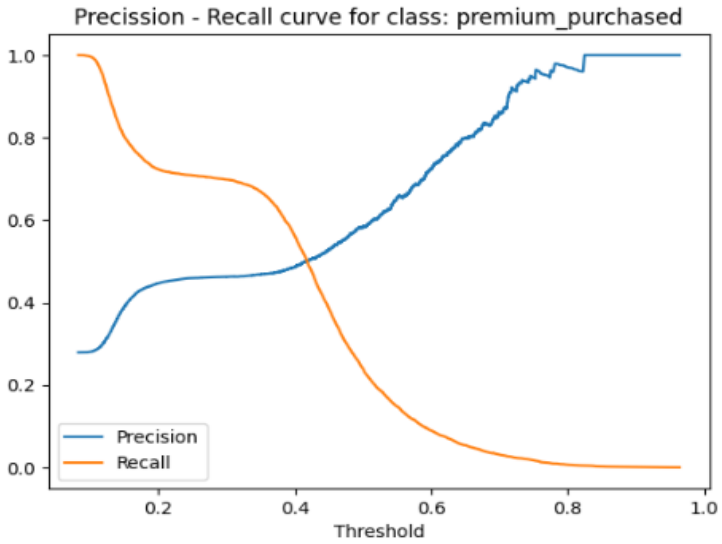
```
Skuteczność w przewidywaniu klasy premium_purchased: 0.5841
Skuteczność w przewidywaniu klasy premium_purchased_this_month: 0.9457
Skuteczność w przewidywaniu klasy premium_purchased_next_month: 0.9687
```

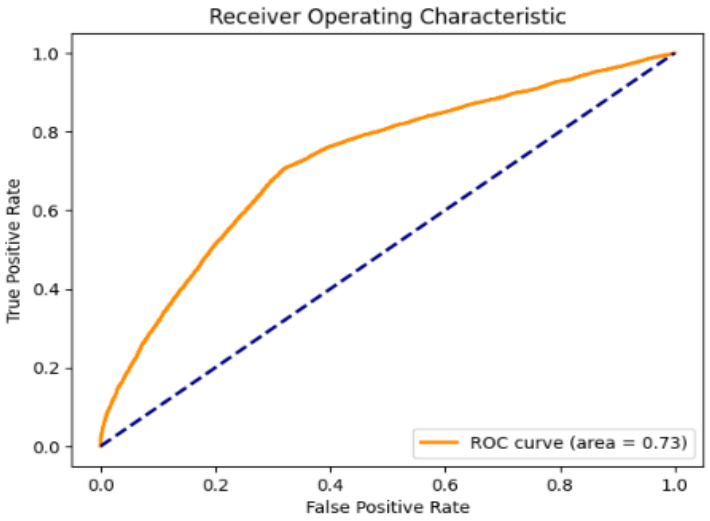
   [TP, FN]
   [FP, TN]

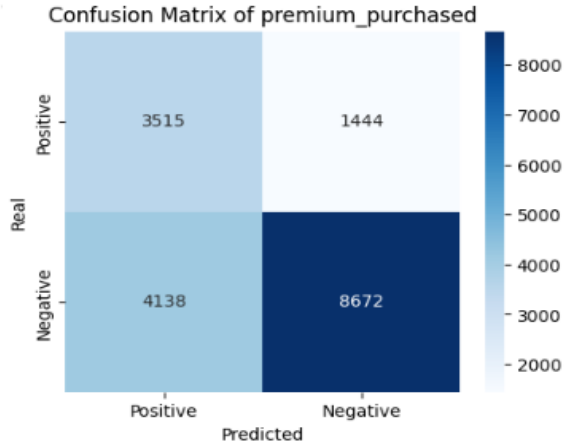
premium_purchased	premium_purchased_this_month	premium_purchased_next_month
[0, 12241] [0, 17192]	[0, 1599] [0, 27834]	[0, 921] [0, 28512]

   Naszym priorytetem i zarazem kryterium sukcesu będzie poprawa tych wyników, koncentrując się szczególnie na zwiększeniu TP (czyli poprawnym klasyfikowaniu użytkowników kupujących premium) oraz zmniejszeniu liczby FN (czyli błędnej klasyfikacji, że użytkownik który kupi premium według modelu go nie kupi).
   Za analityczne kryterium sukcesu posłuży analiza krzywej ROC oraz miara F1. Będziemy dążyć do uzyskania jak najlepszych wartości, ale za minimum sukcesu określamy F1 lepsze niż 0,5 oraz krzywą ROC lepszą od linii przechodzącej przez punkty (0,0) oraz (1,1).

3. **Zaimplementowane modele**
    a) **Regresja logistyczna**
    Prosta regresja logistyczna z bilbioteki sklearn. Oddzielny model dla przewidywania klasy *premium_purchased* i *premium_purchased_this_month*.
    b) **Sieć neuronowa**
    Skorzystaliśmy z biblioteki tensorflow. Jeden model dla przewidywania klasy *premium_purchased* i *premium_purchased_this_month*.
4. **Wyniki**
W drodze prac nad modelami zauważyliśmy, że nie radzą sobie z przewidywaniem klasy *premium_purchased_next_month*. Dostawaliśmy 0 przypadków TP (czyli nigdy modelowi nie udawało się przewidzieć, że dany użytkownik kupi premium w następnym miesiącu). Zrezygnowaliśmy więc z dalszego przewidywania tego atrybutu. Skupiliśmy się na przewidywaniu *premium_purchased* oraz *premium_purchased_this_month*.
    a) **Regresja logistyczna**
        i) **premium_purchased**

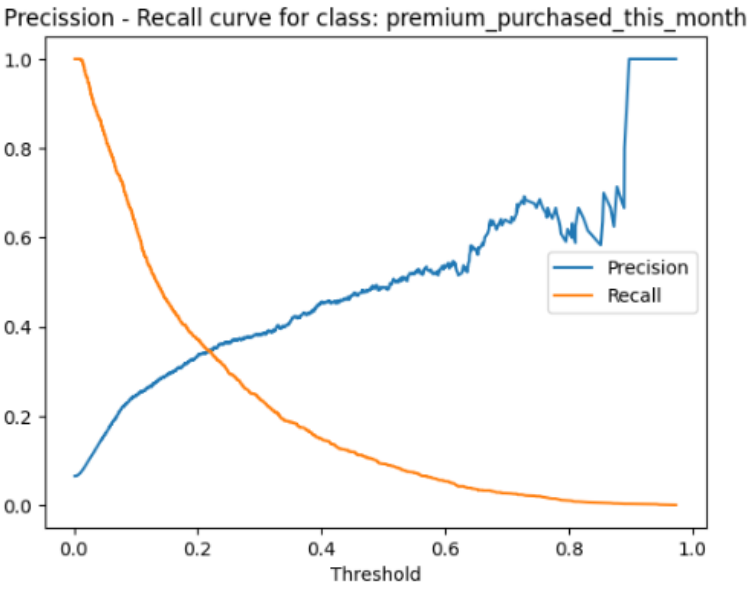


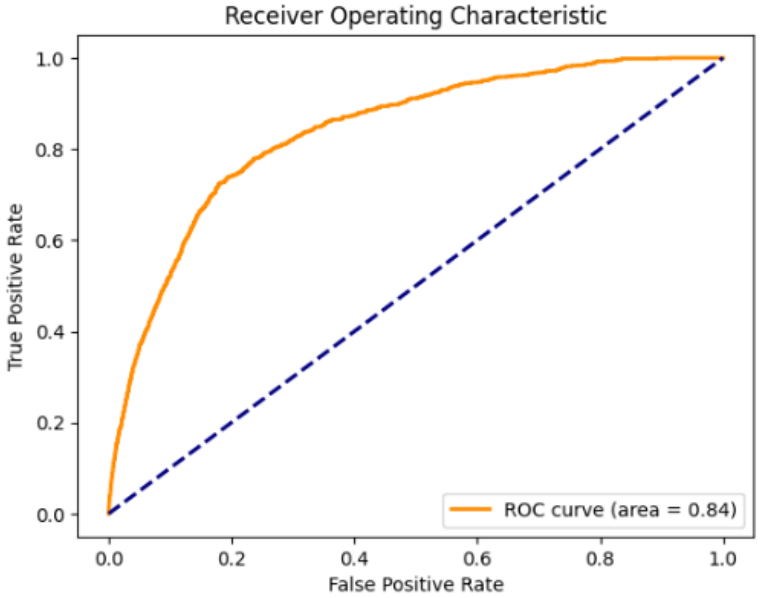


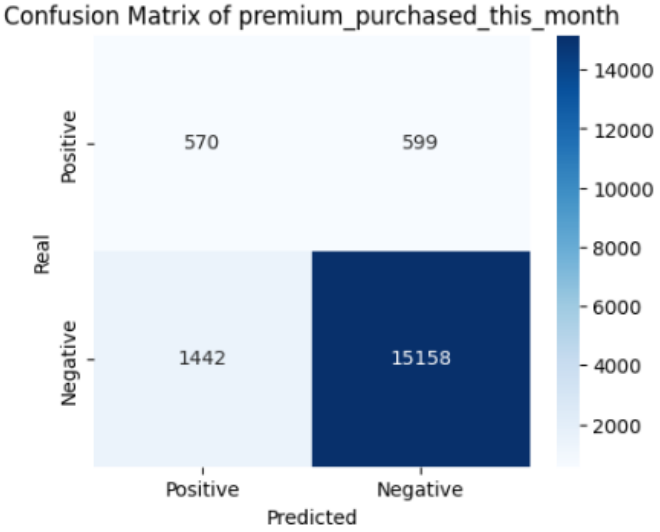



```
Class: premium_purchased
Accuracy: 0.6858573920873431
AUC: 0.6928916884258887
F1-score: 0.5574056454170632
Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.68      0.76     12810
         1.0       0.46      0.71      0.56      4959

    accuracy                           0.69     17769
   macro avg       0.66      0.69      0.66     17769
weighted avg       0.75      0.69      0.70     17769
```

ii) premium_purchased_this_month



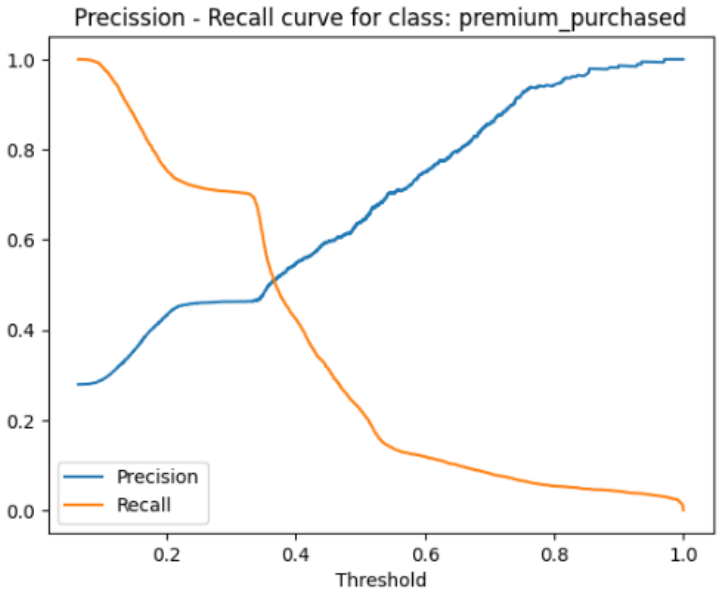


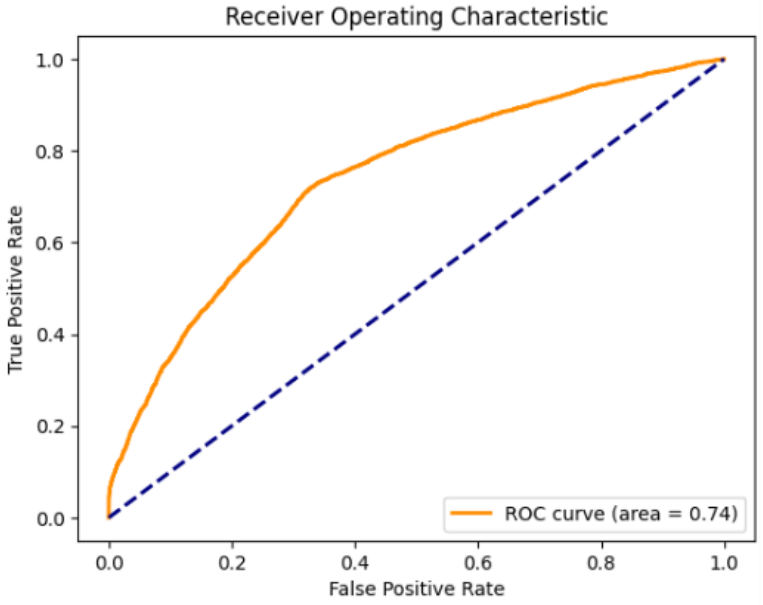


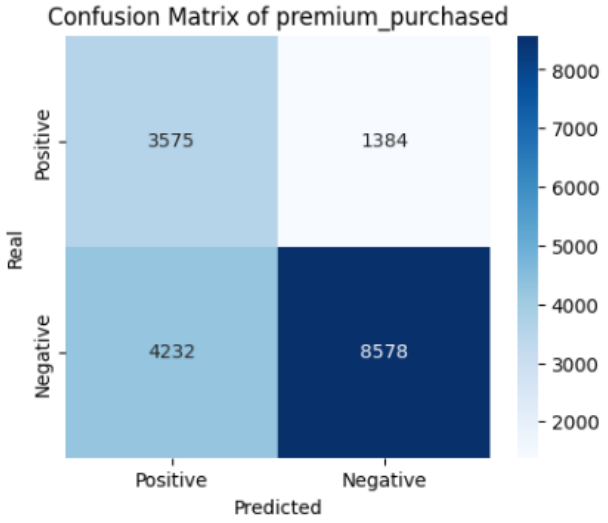

```
Class: premium_purchased_this_month
Accuracy: 0.8851370364117283
AUC: 0.7003643831098559
F1-score: 0.3583778685947815
Classification Report:
              precision    recall  f1-score   support

         0.0       0.96      0.91      0.94     16600
         1.0       0.28      0.49      0.36      1169

    accuracy                           0.89     17769
   macro avg       0.62      0.70      0.65     17769
weighted avg       0.92      0.89      0.90     17769
```

b) Sieć neuronowa

i) premium_purchased

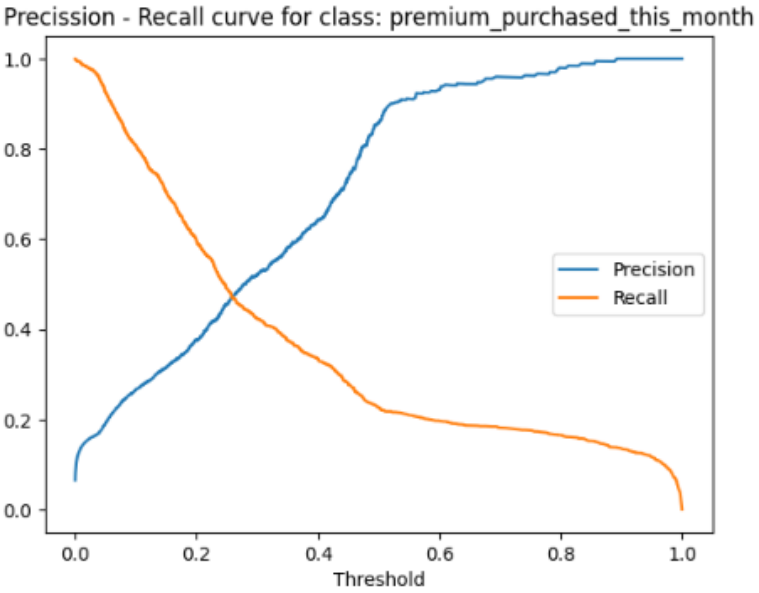


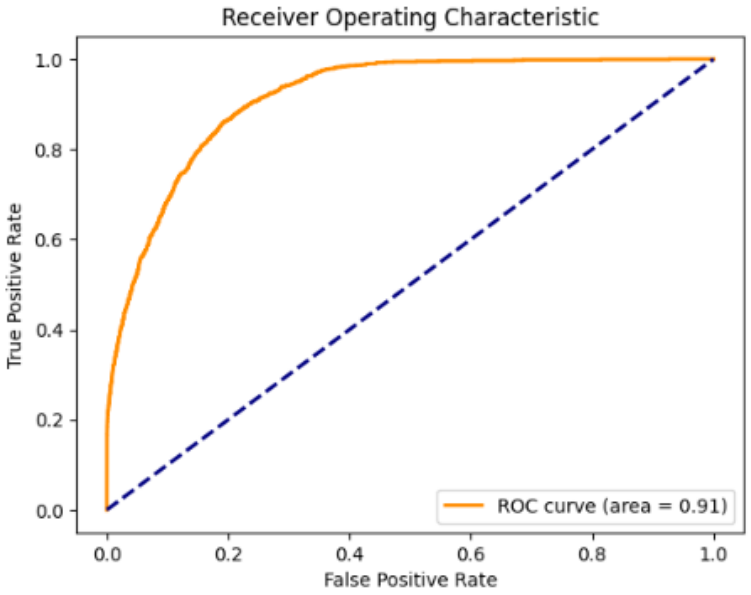


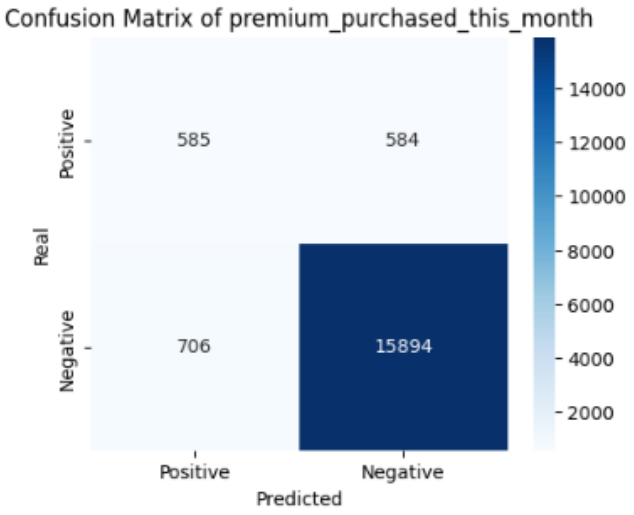



```
Class: premium_purchased
Accuracy: 0.6839439473239912
AUC: 0.6952722866144069
F1-score: 0.560081466395112
Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.67      0.75     12810
         1.0       0.46      0.72      0.56      4959

    accuracy                           0.68     17769
   macro avg       0.66      0.70      0.66     17769
weighted avg       0.75      0.68      0.70     17769
```

ii) premium_purchased_this_month








```
Class: premium_purchased_this_month
Accuracy: 0.9274016545669425
AUC: 0.7289487977573252
F1-score: 0.475609756097561
Classification Report:
              precision    recall  f1-score   support

         0.0       0.96      0.96      0.96     16600
         1.0       0.45      0.50      0.48      1169

    accuracy                           0.93     17769
   macro avg       0.71      0.73      0.72     17769
weighted avg       0.93      0.93      0.93     17769
```

5. **Porównanie modeli**
   Analizując zastosowane modele możemy dojść do wniosku, że lepiej spisuje się model sieci neuronowej. Wykazuje on lepsze wartości miary F1 oraz lepsze krzywe ROC, które to są naszymi analitycznymi kryteriami sukcesu.

F1 SCORE	Regresja logistyczna	**Sieć neuronowa**
premium_purchased	0,5574	**0,5601**
premium_purchased_this_month	0,3584	**0,4756**

POLE POD KRZYWĄ ROC	Regresja logistyczna	**Sieć neuronowa**
premium_purchased	0,73	**0,74**
premium_purchased_this_month	0,84	**0,91**

ACCURACY	Regresja logistyczna	Sieć neuronowa	Naiwny klasyfikator
premium_purchased	**0,6859**	0,6839	0,5841
premium_purchased_this_month	0,8851	0,9274	**0,9457**

   Dokładność dla atrybutu *premium_purchased* jest lepsza dla regresji logistycznej, ale sieć neuronowa ma niewiele mniejszą. Obie metody są lepsze niż naiwny klasyfikator. Z kolei dla atrybutu *premium_purchased_this_month* najlepiej wypada naiwny klasyfikator - trzeba to jednak traktować z przymrużeniem oka ze względu na zdecydowanie większościową klasę oznaczającą w tym przypadku brak zakupu premium w tym miesiącu.

   Ogólnie, biorąc pod uwagę najistotniejsze dla nas miary jakości, **najlepiej wypada sieć neuronowa**.

6. **Mikroserwis - predykcja przy użyciu sieci neuronowej oraz regresji logistycznej**
   Działa przy użyciu uvicorn na ip *127.0.0.1:8000/predict/<lr/nn>*.
   Przesyłamy POSTem atrybuty danego usera. Otrzymujemy odpowiedź:
   - czy user kupi premium?
   - czy user kupi premium w tym miesiącu?

Przykładowe działanie:

sieć neuronowa:

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict/nn' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "number_of_advertisements": 10,
  "number_of_skips": 5,
  "number_of_likes": 7,
  "total_tracks_duration_ms": 1800000,
  "number_of_different_artists": 2,
  "average_release_date": 0.111,
  "average_duration_ms": 1000000,
  "explicit_tracks_ratio": 0.18,
  "average_popularity": 50.0,
  "average_acousticness": 0.3,
  "average_danceability": 0.6,
  "average_energy": 0.5,
  "average_instrumentalness": 0.1,
  "average_liveness": 0.2,
  "average_loudness": -10.3,
  "average_speechiness": 0.1,
  "average_tempo": 90.0,
  "average_valence": 0.4
}'
```

Request URL

```
http://127.0.0.1:8000/predict/nn
```

Server response

Code	Details
200	**Response body**

```
{
  "premium_purchased": 1,
  "premium_purchased_this_month": 0
}
```

Response headers

```
content-length: 56
content-type: application/json
date: Thu,04 Jan 2024 16:59:39 GMT
server: uvicorn
```

Regresja logistyczna

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict/lr' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "number_of_advertisements": 10,
  "number_of_skips": 5,
  "number_of_likes": 7,
  "total_tracks_duration_ms": 1800000,
  "number_of_different_artists": 2,
  "average_release_date": 0.111,
  "average_duration_ms": 1000000,
  "explicit_tracks_ratio": 0.18,
  "average_popularity": 50.0,
  "average_acousticness": 0.3,
  "average_danceability": 0.6,
  "average_energy": 0.5,
  "average_instrumentalness": 0.1,
  "average_liveness": 0.2,
  "average_loudness": -10.3,
  "average_speechiness": 0.1,
  "average_tempo": 90.0,
  "average_valence": 0.4
}'
```

Request URL

```
http://127.0.0.1:8000/predict/lr
```

Server response

Code	Details

200	**Response body**
	`{ "premium_purchased": 0, "premium_purchased_this_month": 0 }`
	Response headers
	`content-length: 56`
	`content-type: application/json`
	`date: Thu,04 Jan 2024 17:01:00 GMT`
	`server: uvicorn`

7. **Mikroserwis - eksperymenty A/B**

   W celu przeprowadzenia eksperymentów A/B wytrenowaliśmy modele danymi do lipca. Późniejsze dane posłużą do porównania skuteczności modeli.
   Podzieliliśmy późniejsze dane na dwie losowo wybrane grupy (A - sieć neuronowa i B - regresja logistyczna) i wrzuciliśmy do modeli.
   Otrzymane wyniki dla poszczególnych grup wyeksportowaliśmy do pliku csv, a następnie skorzystaliśmy z funkcji *ttest_ind(gr_A, gr_B)* z biblioteki scipy.
   Otrzymaliśmy następujące rezultaty:
   dla *premium_purchased:*

```
T-statistic: 29.0460362096324
P-value: 8.168457769831791e-184
Odrzucamy hipotezę zerową. Istnieje istotna różnica między grupami.
Grupa A jest lepsza niż grupa B.
```

   dla *premium_purchased_this_month:*

```
T-statistic: 25.562040790769686
P-value: 1.0377275487561015e-142
Odrzucamy hipotezę zerową. Istnieje istotna różnica między grupami.
Grupa A jest lepsza niż grupa B.
```

8. **Wnioski końcowe**

a) *premium_purchased* vs *premium_purchased_this_month*

   Nasze modele zdecydowanie lepiej radzą sobie z przewidywaniem parametru *premium_purchased* niż *premium_purchased_this_month*. Wynika to z natury dostępnych danych oraz można było się tego spodziewać zwracając uwagę na macierz korelacji. Rekomendujemy więc przykładanie większej uwagi do klasyfikacji parametru *premium_purchased*.

b) Preferowany Model

   Na podstawie analizy wyników modeli, wyraźnie widać, że sieć neuronowa osiągnęła lepsze rezultaty niż regresja logistyczna. Model ten charakteryzuje się wyższymi wartościami miary F1 i krzywej ROC, co świadczy o jego lepszej zdolności do identyfikacji potencjalnych klientów skłonnych do zakupu kont premium.

c) Miary Jakości

   Zarówno dla modelu opartego na sieci neuronowej jak i modelu opartego na regresji logistycznej udało nam się poprawić rezultaty uzyskane dla modelu naiwnego. Krzywa ROC wygląda zdecydowanie lepiej niż prosta linia.

d) Eksperymenty A/B

   Testy A/B potwierdziły istotne statystycznie różnice między grupą korzystającą z sieci neuronowej a grupą korzystającą z regresji logistycznej. Sieć neuronowa okazała się bardziej efektywna w identyfikacji użytkowników zainteresowanych zakupem kont premium.

Podsumowując, na podstawie analizy i wyników uzyskanych z różnych aspektów projektu, rekomendujemy skoncentrowanie się na implementacji sieci neuronowej jako preferowanego modelu dla prognozowania skłonności użytkowników do zakupu kont premium. Wdrożenie tej rekomendacji może przynieść korzyści biznesowe poprzez lepsze zrozumienie zachowań klientów i efektywniejsze działania marketingowe.