Uniwersytet Gdański
Wydział Matematyki, Fizyki i Informatyki
Instytut Informatyki

Analiza danych użytkowników social media

Maria Koren

Gdańsk

12 maja 2024

Spis treści

1 Wstęp

Została przeprowadzona analiza danych użytkowników social media na podstawie bazy danych link.W celu analizy najpierw wykonano preprocessing danych, klasyfikacja prostymi klasyfikatorami, klasyfikacja sieciami neuronowymi. Oraz wyszukano reguły asocjacyjne.

2 Preprocessing

2.1 Znaczenia kolumn, problemy w bazie

Główne zadania preprocessingu umieszczone w pliku *main.py*.

Baza danych *SocialMediaUsersDataset.csv* zawiera następujące kolumny: UserID, Name, Gender, DOB, Interests, City, Country.

Dla dalszej analizy kolumny UserId oraz Name usunięte, ponieważ nie wnoszą żadnych informacji. I została zapisana do pliku *DataWithNoNameNoId.csv*.

Następnie kolumne DOB, przechowująca daty urodzenia użytkowników przerobiono na kolumne Age, gdzie jest wiek.

Zatem kolumnę Gender, która przechowuje dane o płci w postaci słów "Male"oraz "Female"zmapowano na 0 oraz 1 w następujący sposób:

```
01 | gender_mapping = {'Male': 0, 'Female': 1}
02 | df['Gender'] = df['Gender'].map(gender_mapping)
```

Jednym z największych problemów w tej bazie są kraje oraz miasta. Ponieważ ilość wystąpień każdego kraju jest dość różna. Na przykład najczęściej się pojawia 'United States' 12311 razy, a najmniej 'Saint Lucia', 'American Samoa', 'Saint Martin' pojawiają się jednen raz. Z miastami jest trudniej, ponieważ one są powiązane z krajami. Dlatego przyjęte było następujące podejście: usunąć kolumnę 'City'.A dla krajów wybrano 10 z nich najczęście występujących.

Drugim dużym problemem w bazie była kolumna 'Interests', reprezentująca zainteresowania osób. Problemem w tej kolumnie jest to, że może ona zawierać w sobie jak zarówno jedno zainteresowanie, jak i kilka z nich. Również występował problem, że zaiteresowanie 'Fashion' było blisko 18 000 razy, a reszta około 9 000. Więc do finalnej bazy wybrano 9 najczęstszych z wyjątkiem 'Fashion'.

Z uwagi na małą ilość kolumn w bazie dodano jeszcze kolumnę InterestCount, reprezentującą ilość zainteresowań danej osoby.

2.2 Finalna Baza

Finalna baza zawiera w sobie kolumny tylko liczbowe, co ułatwia przetwarzanie danych. Kolumny to: "Gender, Interests, Country, InterestCount, Age". Każde zainteresowanie oraz kraj zostały zmapowane na liczby. Przy czym dla każdego zainteresowania występuje równa ilość rekordów dla każdego kraju, zrobiono to w celu zbalansowania bazy danych. Takie zestawienie bazy danych daje łącznie 6 300 rekordów w bazie *data.csv*.

```
selected_countries = ['United States', 'India', 'China', 'Brazil', '
    Russia', 'Germany', 'Japan', 'United Kingdom', 'France', 'Mexico']
selected_interests = ["'Cooking'", "'Pets'" ,"'Movies'" ,"'Gaming'", "'
    Fitness'" ,"'Outdoor activities'", "'Travel'", "'Business and
    entrepreneurship'" , "'Social causes and activism'"]

country_to_index = {country: index for index, country in enumerate(
    selected_countries)}
interest_to_index = {interest: index for index, interest in enumerate(
    selected_interests)}

df['Country'] = df['Country'].apply(lambda x: country_to_index[x] if x
    in selected_countries else None)
df = df.dropna(subset=['Country']).astype({'Country': 'int'})
df = df[df['Interests'].isin(selected_interests)]
df['Interests'] = df['Interests'].apply(lambda x: interest_to_index[x]
    if x in selected_interests else None)
df = df.dropna(subset=['Interests']).astype({'Interests': 'int'})

final_df = pd.DataFrame(columns=df.columns)
grouped = df.groupby(['Country', 'Interests'])
for (country, interest), data in grouped:
    if len(data) < 70:
        final_df = pd.concat([final_df, data], ignore_index=True)
    else:
        sampled_data = data.sample(n=70, random_state=42)
        final_df = pd.concat([final_df, sampled_data], ignore_index=True
    )

final_df = final_df.groupby(['Country', 'Interests']).head(70).
    reset_index(drop=True)

final_df.to_csv('data.csv', index=False)
```

Również jako preprocessing wykonano skalowanie danych w kolumnach InterestCount i Age, plik *scaled.py*. Skalowana baza danych jest zapisana do pliku *data_ scaled.csv*

3 Klasyfikacja prostymi klasyfikatorami

3.1 Klasyfikacja zainteresowań

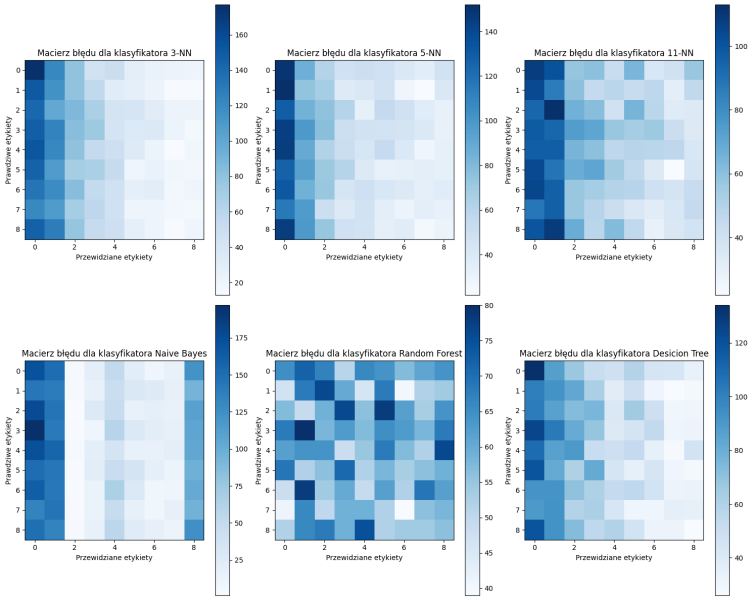
Wykonano klasyfikację prostymi klasyfikatorami poznanymi na zajęciach:

1. Nearest Neighbor (3, 5, 11, 100)
2. Naive Bayes
3. Desicion Tree

Oraz na klasyfikatorze, znalezionym samodzielnie: Random Forest (łączy w sobie kilka drzew decyzyjnych).

Za pomocą wyżej wymienionych klasyfikatorów najpierw klasyfikałam zainteresowania. Na bazie danych bez skalowania wyszły takie wyniki:

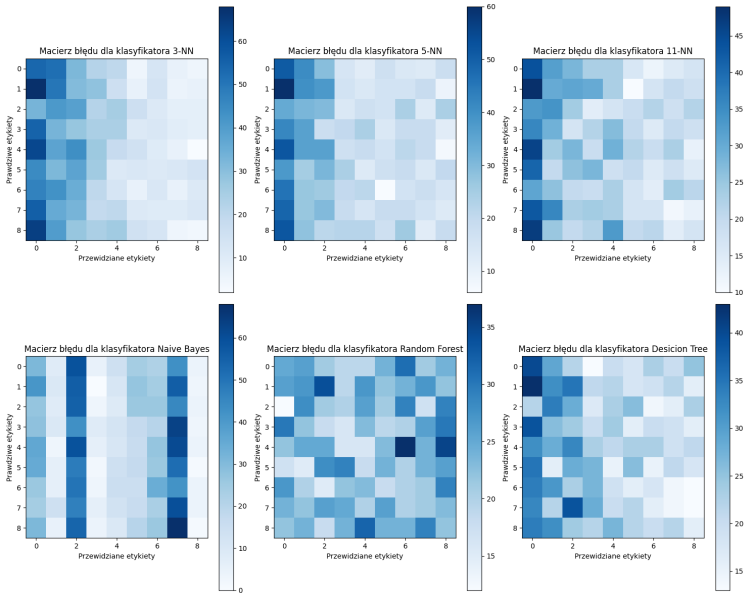
- Dokładność klasyfikatora 3-NN: 0.12163839470417874
- Dokładność klasyfikatora 5-NN: 0.10881257757550683
- Dokładność klasyfikatora 11-NN: 0.11170872983036823
- Dokładność klasyfikatora 100-NN: 0.11170872983036823
- Dokładność klasyfikatora Naive Bayes: 0.1119155978485726
- Dokładność klasyfikatora Random Forest: 0.11005378568473315
- Dokładność klasyfikatora Desicion Tree: 0.11605295821266032



Rysunek 1: Proste klasyfikatore na bazie bez skalowania

Na bazie ze skalowaniem wyniki wyszły następujące:

- Dokładność klasyfikatora 3-NN: 0.12246586677699628
- Dokładność klasyfikatora 5-NN: 0.11708729830368225
- Dokładność klasyfikatora 11-NN: 0.112122465866777
- Dokładność klasyfikatora 100-NN: 0.112122465866777
- Dokładność klasyfikatora Naive Bayes: 0.1119155978485726
- Dokładność klasyfikatora Random Forest: 0.10819197352089367
- Dokładność klasyfikatora Desicion Tree: 0.11563922217625155



Rysunek 2: Proste klasyfikatore na bazie ze skalowaniem

Dla klasyfikatorów najbliższych sąsiadów lepsze wyniki dała baza ze skalowaniem, natomiast dla Random Forest oraz Desicion Tree baza bez skalowania daje lepsze wyniki. W każdym z przypadków lepsze wyniki dał klasyfikator 3NN.

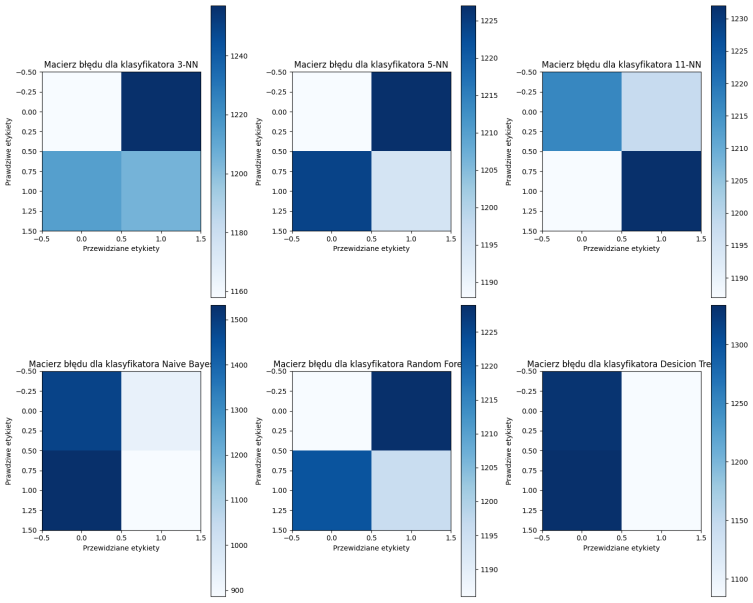
Podsumowanie. Ponieważ jest 9 klas, prawdopodobieństwo losowego trafiania w poprawne zainteresowanie wynosi 0.11, można wnioskować że te klasyfikatory poradziły sobie trochę lepiej niż zwykłe losowanie.

3.2 Klasyfikacja płci

Kolejną rzeczą którą sklasyfikowałam za pomocą prostych klasyfikatorów jest klasyfikacja płci.

Na bazie danych bez skalowania wyszły takie wyniki:

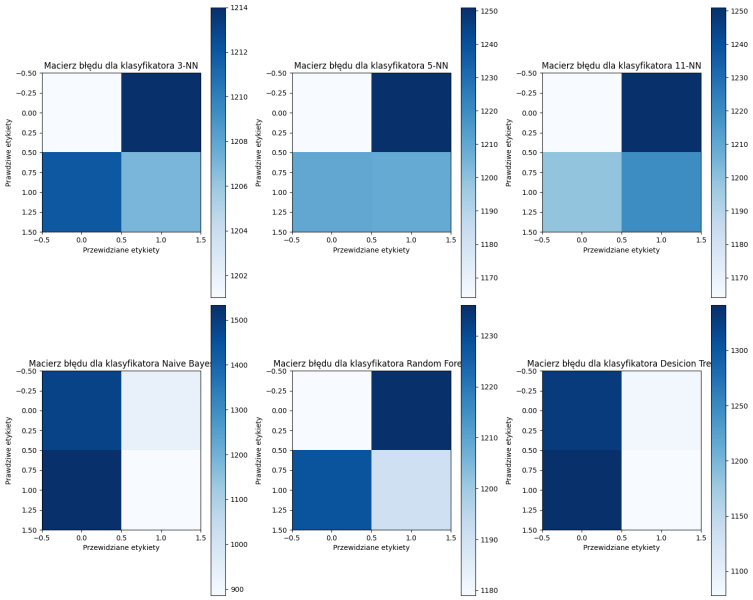
- Dokładność klasyfikatora 3-NN: 0.4888291270169632
- Dokładność klasyfikatora 5-NN: 0.49296648738105087
- Dokładność klasyfikatora 11-NN: 0.5066197765825403
- Dokładność klasyfikatora 100-NN: 0.5066197765825403
- Dokładność klasyfikatora Naive Bayes: 0.49007033512618947
- Dokładność klasyfikatora Random Forest: 0.4927596193628465
- Dokładność klasyfikatora Desicion Tree: 0.49958626396359124



Rysunek 3: Proste klasyfikatore na bazie bez skalowania

Na bazie ze skalowaniem wyniki wyszły następujące:

- Dokładność klasyfikatora 3-NN: 0.49813818783616054
- Dokładność klasyfikatora 5-NN: 0.49089780719900705
- Dokładność klasyfikatora 11-NN: 0.4931733553992553
- Dokładność klasyfikatora 100-NN: 0.4931733553992553
- Dokładność klasyfikatora Naive Bayes: 0.49007033512618947
- Dokładność klasyfikatora Random Forest: 0.4902772031443939



Rysunek 4: Proste klasyfikatore na bazie ze skalowaniem

- Dokładność klasyfikatora Desicion Tree: 0.49793131981795613

Wnioski odnośnie skalowania danych takie, jak poprzednio.

Podsumowanie. Ponieważ klas jest tylko 2, to prawdopodobieństwo losowego trafiania w poprawne zainteresowanie wynosi 0.5. Więc można wnioskować, że proste klasyfikatory w tym przykładzie poradziły sobie gorzej niż losowanie. Trochę lepsze wyniki dali 11NN oraz 100NN na bazie bez skalowania.

4 Klasyfikacja sieciami neuronowymi

Za pomocą sieci neuronowych sklasyfikowano 2 opcje:

1. Klasyfikacja zainteresowań (plik *nn.py*)
2. Klasyfikacja krajów (plik *nn2.py*)

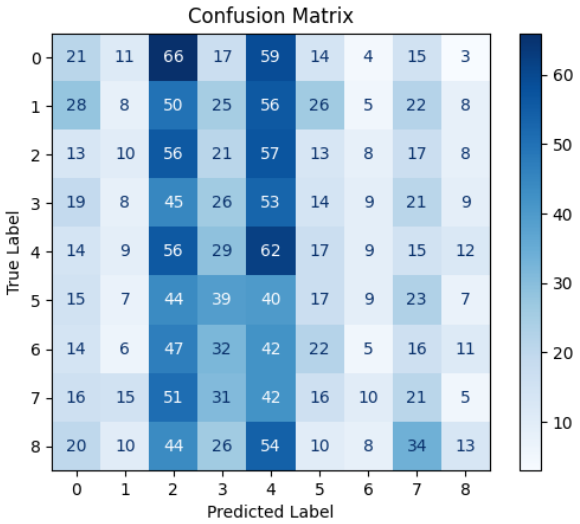
4.1 Klasyfikacja zainteresowań

W wersji ostatecznej została użyta taka sieć:

```
model = Sequential([
    Dense(64, activation='tanh', input_shape=(X_train.shape[1],)),
    Dense(32, activation='tanh'),
    Dense(len(np.unique(y_train)), activation='softmax')
])
```

Przed zostawieniem tego modelu były użyte inne sieci, zmieniono ilość warstw, funkcję aktywacyjną. Ale najlepsze wyniki, choć nie o wiele, pokazała właśnie ta sieć. Sieć została wytrenowana na 1000 epokach.

Dokładność tej sieci: $12,12\%$.



Rysunek 5: Macierz błedów

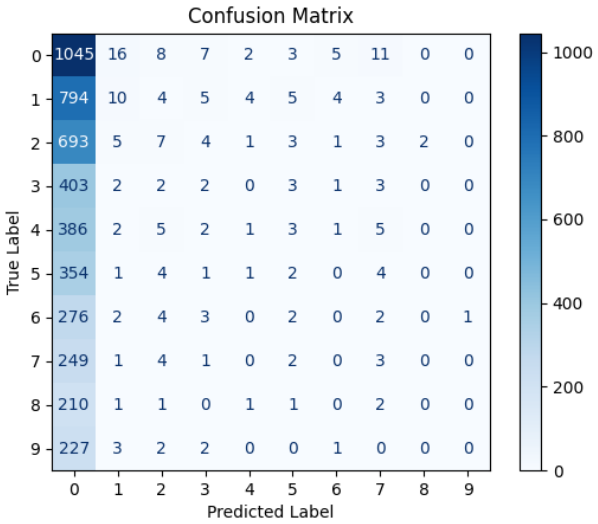
4.2 Klasyfikacja krajów

W wersji ostatecznej została użyta taka sieć:

```
model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(len(np.unique(y_train)), activation='softmax')
])
```

Przed zostawieniem tego modelu były użyte inne sieci, zmieniono ilość warstw i funkcji aktywacyjne. Ale najlepsze wyniki, choć nie o wiele pokazała właśnie ta sieć. Sieć została wytrenowana na 5000 epokach.

Dokładność tej sieci: $10,01\%$.



Rysunek 6: Macierz błedów

5 Reguły asocjacyjne

5.1 Reguły zainteresowań

Reguły asocjacyjne zostały zbadane odnośnie zainteresowań. Najpierw na pliku oryginalnym zrobiony był dataframe, gdzie kolumny to unikalne zainteresowania, a w wierszch stoją 0 jeżeli zainteresowanie nie wystepuje u danej osoby, 1 jeżeli występuje. Na danych oryginalnych reguły w skrócie można było reprezentować w ten sposób: "Jeżeli interesujesz się czym kolwiek, prawdopodobnie będzisz się interesował Fashion". To wychodziło z powodu mocno niezbalansowanej bazy, o czym było napisane w części o preprocessingu.

Zatem wykonano inny preprocessing: usunięto wszystkie rekordy, zawierające 'Fashion', i powórzono eksperyment. Wyszła 1 reguła:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	('Sports')	('Cooking')	0.097022	0.099975	0.009914	0.102185	1.022102	0.000214	1.002461	0.023948

Rysunek 7: Reguła zainteresowań

Reguła ta oznacza: "Jeżeli interesujesz się sportem, z prawdopodobieństwem 0.097 będziesz się interesował gotowaniem".

5.2 Inne reguły

Ponieważ funkcji reguł asocjacyjnych wymagają tylko wartości 0/1 bądź True/False, dla pliku *data.csv* jest potrzebny dalszy preprocessing. W nim usunięto kolumny Interests oraz Country. Dla kolumn Age oraz InterestCount wyliczono średnie znaczenia, wartości poniżej średniej zamieniono na 0, powyżej na 1. Średnia dla wieku jest 44,33, dla zainteresowań 3. Dla przypomnienia, w kolumnie Gender: Male - 0, Female - 1. Wyszły takie reguły:

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
(InterestCount)	(Age)	0.406796	0.498253	0.208638	0.512881	1.029357	0.005950	1.030028	0.048077
(Age)	(Gender)	0.498253	0.500794	0.252779	0.507330	1.013050	0.003256	1.013266	0.025675
(Gender)	(Age)	0.500794	0.498253	0.252779	0.504756	1.013050	0.003256	1.013130	0.025806
(InterestCount)	(Gender)	0.406796	0.500794	0.202763	0.498439	0.995297	-0.000958	0.995304	-0.007902
(Age)	(InterestCount)	0.498253	0.406796	0.208638	0.418738	1.029357	0.005950	1.020545	0.056841
(Gender)	(InterestCount)	0.500794	0.406796	0.202763	0.404883	0.995297	-0.000958	0.996785	-0.009377

Rysunek 8: Inne reguły

Opis reguł:

1. Jeżeli masz powyżej 3 zainteresowań, raczej jesteś w wieku powyżej 44 lat
2. Jeżeli jesteś w wieku powyżej 44 lat, raczej jesteś kobietą.
3. Jeżeli jesteś kobietą, raczej masz powyżej 44 lat.
4. Jeżeli masz powyżesz 3 zainteresowań, raczej jesteś kobietą.
5. Jeżeli masz powużej 44 lat, raczej masz więcej trzech zainteresowań.
6. Jeżeli jesteś kobietą, raczej masz powyżej 3 zainteresowań.

6 Podsumowanie

W pracy przeprowadzono analizę danych użytkowników social media. Skorzystano zarówno s prostych klasyfikatorów, jak i sieci neuronowych, będących bardziej złożonym klasyfikatorem.

Fakt, że wynik każdego z klasyfikatoru jest blisko prawdobodobieństwa losowego "strzelania"można wytłumaczyć tym, że analizowane dane mają słabe zależności. To oznacza, że w rzeczywistym świecie trudno zrobić predykcję o zainteresowaniach, na podstawie płci, wieku, miejsca zamieszkania, co zmniejsza szanse na dyskryminację. Również trudno na podstawie zainteresowań, płci i wieku wywnioskować o miejscu zamieszkania, co dodatkowo zmniejsza szansę na dyskryminację.