|  |  |
| --- | --- |
| Projekt – sieci neuronowe | Data złożenia projektu: 07.05.2021 |
| Numer grupy projektowej: 60 | Imię i nazwisko I: Szymon Karpęcki  Imię i nazwisko II: Szymon Lorenc |

Przewidywanie ceny za ofertę Airbnb w Nowym Jork

# Opis problemu i danych

Wykorzystany zbiór danych dotyczy ofert dostępnych w Airbnb dla obszaru obejmującego miasto Nowy Jork. Pierwotnie zbiór zawierał następujące kolumny:

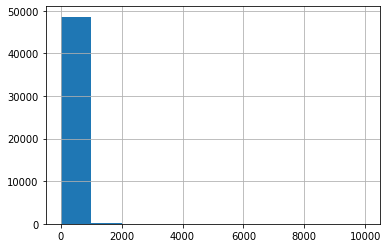
Tabela 1

|  |  |  |
| --- | --- | --- |
| **Nazwa** | **Rodzaj** | **Usunięta przed uczeniem modelu** |
| id | Ilościowa | Tak (nie daje wartościowej informacji) |
| name | Jakościowa | Tak (nie daje wartościowej informacji) |
| host\_id | Ilościowa | Tak (nie daje wartościowej informacji) |
| host\_name | Jakościowa | Tak (nie daje wartościowej informacji) |
| neighbourhood\_group | Jakościowa | Nie |
| neighbourhood | Jakościowa | Nie |
| latitude | Ilościowa | Nie |
| longitude | Ilościowa | Nie |
| room\_type | Jakościowa | Nie |
| price | Ilościowa | Nie |
| minimum\_nights | Ilościowa | Nie |
| number\_of\_reviews | Ilościowa | Nie |
| last\_review | Jakościowa | Tak (posiada wiele pustych, trudnych do zastąpienia wartości) |
| reviews\_per\_month | Ilościowa | Nie |
| calculated\_host\_listings\_count | Ilościowa | Tak (nie daje wartościowej informacji) |
| availability\_365 | Ilościowa | Nie |

Ponadto obliczono podstawowe statystki dla zmiennych ilościowych:

Tabela 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **latitude** | **longitude** | **price** | **minimum\_nights** | **number\_of\_reviews** | **reviews\_per\_month** | **availability\_365** |
| **count** | 48895 | 48895 | 48895 | 48895 | 48895 | 48895 | 48895 |
| **mean** | 40.72895 | -73.95217 | 152.72069 | 7.029962 | 23.274466 | 1.09091 | 112.781327 |
| **std** | 0.05453 | 0.046157 | 240.15417 | 20.51055 | 44.550582 | 1.597283 | 131.622289 |
| **min** | 40.49979 | -74.24442 | 0 | 1 | 0 | 0 | 0 |
| **max** | 40.91306 | -73.71299 | 10000 | 1250 | 629 | 58.5 | 365 |

Histogram oraz ilość wartości w poszczególnych przedziałach dla zmiennej wyjściowej **price** prezentuje się następująco:

Rysunek 1

Tabela 3

|  |  |  |
| --- | --- | --- |
| Zakres | Ilość wartości | % Wszystkich wartości |
| (-0.001, 100.0] | 23928 | 48.94% |
| (100.0, 200.0] | 16583 | 33.92% |
| (200.0, 300.0] | 5027 | 10.28% |
| (300.0, 400.0] | 1594 | 3.26% |
| (400.0, 500.0] | 719 | 1.47% |
| (500.0, 1000.0] | 805 | 1.65% |
| (1000.0, 10000.0] | 239 | 0.49% |

Można zauważyć, że w przedziale cenowym od 0 do 300 USD znajduje się ponad 93% wszystkich wartości. Wynika to ze specyfiki Airbnb, gdzie znaczną większość stanowią oferty wynajmu relatywnie niewielkich pomieszczeń za niską kwotę. Z tego powodu postanowiono usunąć ze zbioru oferty z kwotami wynajmu wyższymi niż 300 USD – miało to znaczny wpływ na dokładność przewidywań modelu (niemal trzykrotny wzrost R^2). Statystyki cen po wyeliminowaniu wartości przekraczających 300 USD oraz równych 0 wyglądają następująco:

|  |  |
| --- | --- |
|  | **price** |
| **count** | 45527 |
| **mean** | 118.31265 |
| **std** | 65.89067 |
| **min** | 10 |
| **max** | 300 |

Korelacje między zmienną wyjściową **price,** a zmiennymi wejściowymi są następujące:

|  |  |
| --- | --- |
|  | **price** |
| **nghb\_group** | -0.023196 |
| **nghb** | -0.142119 |
| **latitude** | 0.067296 |
| **longitude** | -0.305698 |
| **room\_type** | 0.489204 |
| **price** | 1 |
| **minimum\_nights** | 0.028076 |
| **number\_of\_reviews** | -0.027798 |
| **reviews\_per\_month** | -0.05022 |
| **availability\_365** | 0.057972 |

Jedyna zmienne, które są skorelowane ze zmienną wyjściową to:

* **room\_type –** typ pokoju, korelacja umiarkowanie-silnia
* **longitutude** – dł. geograficzna, korelacja umiarkowana

Zadaniem modelu było przewidywanie ceny za mieszkanie w zależności od danych wejściowych, jest to zatem problem **regresji.**

# Obróbka danych

Transformacja danych zawierała następujące kroki:

1. Usunięcie kolumn zgodnie z tabelą 1.
2. Usunięcie wierszy, dla których cena przekraczała 300 USD lub była równa 0 USD.
3. Jedyna kolumna posiadająca wartości NaN to **reviews\_per\_month.** NaN tutaj wynikało z braku recenzji dla danej oferty, zatem zastąpiono je wartością 0.
4. Zmienne jakościowe (tj. neighbourhood, neighbourhood\_group oraz room\_type) przekształcono na wartości numeryczne stosując dostępną w module pandas funkcję factorize() przypisując każdej ww. zmiennej wartości od 0…n, gdzie n to liczba unikalnych kategorii danej zmiennej.
5. Ostatni etap transformacji danych, czyli standaryzacja to również pierwsza warstwa zaimplementowanej sieci. Standaryzację wykonano wykorzystując instancję klasy Normalization.

# Opis zastosowanych sieci neuronowych

Do budowy sieci wykorzystano framework Keras dostępny w module tensorflow dla języka Python. Stworzono głęboką sieć neuronową (DNN – Deep Neural Network). Zdecydowano się na 100 epok uczenia. Funkcją straty dającą najlepsze rezultaty okazała się MEA. Wykorzystane metryki to:

* root\_mean\_squared\_error
* mean\_squared\_error
* mean\_absolute\_error

Są to typowe metryki wykorzystywane w modelach realizujących problem regresji.

Stworzono sieci o następujących architekturach:

**Sieć 1:**

* Warstwa 1:
  + Rodzaj warstwy: wejściowa,
  + Liczba neuronów: 9
* Warstwa 2:
  + Rodzaj warstwy: pośrednia,
  + Liczba neuronów: 500,
  + Funkcja aktywacji: relu
* Warstwa 3:
  + Rodzaj warstwy: pośrednia,
  + Liczba neuronów: 250,
  + Funkcja aktywacji: relu
* Warstwa 4:
  + Rodzaj warstwy: pośrednia,
  + Liczba neuronów: 125,
  + Funkcja aktywacji: relu
* Warstwa 5:
  + Rodzaj warstwy: pośrednia,
  + Liczba neuronów: 62,
  + Funkcja aktywacji: relu
* Warstwa 6:
  + Rodzaj warstwy: wyjściowa,
  + Liczba neuronów: 1,
  + Funkcja aktywacji: n/d

**Sieć 2:**

* Warstwa 1:
  + Rodzaj warstwy: wejściowa,
  + Liczba neuronów: 9
* Warstwa 2:
  + Rodzaj warstwy: pośrednia,
  + Liczba neuronów: 64,
  + Funkcja aktywacji: relu
* Warstwa 3:
  + Rodzaj warstwy: wyjściowa,
  + Liczba neuronów: 1,
  + Funkcja aktywacji: n/d

# Dyskusja wyników oraz wnioski

Wyniki wykonanej ewaluacji modelów:

Tabela 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr. architektury** | **Typ danych** | **MSE** | **RMSE** | **MAE** | **R2** |
| 1 | trenujace | 1703.270142 | 41.2707 | 27.99689 | 0.606598 |
| 1 | testowe | 2000.362671 | 44.72541 | 31.45818 | 0.542939 |
| 2 | trenujace | 2030.783813 | 45.06422 | 31.36546 | 0.530952 |
| 2 | testowe | 2041.399048 | 45.18184 | 31.60384 | 0.533563 |

Mimo różnicy w ilości warstw oraz liczbie neuronów na każdej warstwie sieci dokładność obu modeli dla danych testowych jest podobna. Tym niemniej dla pierwszej architektury model nieco lepiej przewiduje wyniki dla danych trenujących co może świadczyć o przetrenowaniu modelu.

Wyniki przewidywań dla zastosowanych metryk, szczególnie współczynnik determinacji R2, wskazują, że model przewiduje ceny ofert w wystarczającym stopniu, aby w przybliżeniu stwierdzić zakres cenowy w jakim powinna znaleźć się dana oferta.

Dalszy rozwój modelu należałoby oprzeć na modyfikacji liczby warstw, neuronów na poszczególnych warstwach oraz zwiększeniu liczby epok uczenia.