# Analiza Predykcyjna Skłonności do Subskrypcji w Oparciu o Modele Uczenia Maszynowego

**Autorzy:** Artiom Herashchenko K3, Maksym Kudlai K2, Oleksandr Burmahin K3

**Data:** 22 czerwca 2025

**Streszczenie:** Niniejsze badanie koncentruje się na zastosowaniu i ocenie dwóch fundamentalnych algorytmów klasyfikacji – Naiwnego Klasyfikatora Bayes'a oraz Drzewa Decyzyjnego – w celu predykcji skłonności klientów do subskrypcji usługi. Wykorzystując zbiór danych subscribers.xlsx, przeprowadzono kompleksowy proces, od przygotowania danych, przez trening modeli, po ich ewaluację. Celem badania jest nie tylko zbudowanie działających modeli, ale również dogłębne zrozumienie ich działania, porównanie ich skuteczności oraz identyfikacja potencjalnych ograniczeń.

### 1. Wprowadzenie

W dobie cyfryzacji i rosnącej konkurencji, zdolność do przewidywania zachowań klientów jest kluczowa dla sukcesu biznesowego. Uczenie maszynowe dostarcza potężnych narzędzi do analizy danych i tworzenia modeli predykcyjnych. Celem niniejszego projektu badawczego jest zbadanie możliwości przewidzenia, czy klient zdecyduje się na subskrypcję produktu, na podstawie ograniczonych danych demograficznych i finansowych.

W ramach badania, analizie poddano zbiór danych zawierający cztery cechy opisujące klientów (Age, Sex, Income, Residence) oraz jedną zmienną docelową (Subscribes). Do zadania klasyfikacji binarnej wybrano dwa popularne i interpretowalne modele: Naiwny Klasyfikator Bayes'a, oparty na teorii prawdopodobieństwa, oraz Drzewo Decyzyjne, bazujące na logicznych regułach decyzyjnych.

### 2. Metodologia Badawcza

Proces badawczy został przeprowadzony w sposób metodyczny i powtarzalny, z wykorzystaniem języka Python oraz bibliotek pandas i scikit-learn. Poniżej przedstawiono kolejne kroki.

#### Krok 1: Akwizycja i Wstępne Przetwarzanie Danych

* **Źródło Danych:** Badanie oparto na pliku subscribers.xlsx, który został wczytany do środowiska analitycznego przy użyciu biblioteki pandas z silnikiem openpyxl.
* **Czyszczenie Danych:** W pierwszym etapie nazwy kolumn zostały poddane normalizacji poprzez usunięcie ewentualnych białych znaków na początku i na końcu (df.columns.str.strip()), co jest dobrą praktyką zapobiegającą błędom w dalszej części procesu.
* **Identyfikacja Zmiennej Docelowej:** Zmienna Subscribes została zidentyfikowana jako zmienna celu (target). Jej tekstowe wartości ('Yes'/'No') zostały przekonwertowane na format binarny (1/0), który jest standardem w modelach klasyfikacyjnych.
* **Identyfikacja Cech:** Pozostałe kolumny zidentyfikowano jako cechy (features) i podzielono na dwa typy:
  + **Cechy numeryczne:** Income
  + **Cechy kategoryczne:** Age, Sex, Residence

#### Krok 2: Przygotowanie Cech (Feature Engineering & Preprocessing)

Algorytmy uczenia maszynowego wymagają, aby wszystkie dane wejściowe miały format numeryczny. Ponadto, ich skuteczność często zależy od odpowiedniego skalowania cech. W tym celu zaimplementowano potok przetwarzania (Pipeline) z użyciem ColumnTransformer:

* **Skalowanie Cech Numerycznych:** Do kolumny Income zastosowano StandardScaler. Standaryzacja przekształca dane tak, aby miały średnią równą 0 i odchylenie standardowe równe 1. Jest to kluczowe dla algorytmów, które są wrażliwe na skalę cech (np. maszyny wektorów nośnych, regresja logistyczna) i często poprawia wyniki również w przypadku Naiwnego Bayes'a.
* **Kodowanie Cech Kategorycznych:** Do cech Age, Sex, Residence zastosowano OneHotEncoder. Technika ta przekształca każdą kategorię w nową, binarną kolumnę (0 lub 1), co pozwala modelom na pracę z danymi nienumerycznymi bez wprowadzania błędnego założenia o ich porządkowej relacji.

#### Krok 3: Podział Zbioru Danych

Aby rzetelnie ocenić zdolność modelu do generalizacji na nowych, niewidzianych wcześniej danych, zbiór został podzielony na dwie części:

* **Zbiór treningowy (75% danych):** Używany do "nauczenia" modelu.
* **Zbiór testowy (25% danych):** Używany do ostatecznej oceny modelu.

Podziału dokonano za pomocą funkcji train\_test\_split, używając dwóch ważnych parametrów:

* random\_state=42: Zapewnia powtarzalność podziału, co jest kluczowe dla odtwarzalności wyników badań.
* stratify=y: Gwarantuje, że proporcje klas ('Yes'/'No') w zbiorze treningowym i testowym są takie same jak w oryginalnym zbiorze. Jest to niezwykle ważne w przypadku niezbalansowanych zbiorów danych.

#### Krok 4: Budowa i Trening Modeli

Zdefiniowano dwa odrębne potoki (Pipeline), z których każdy łączył kroki przetwarzania wstępnego z finalnym klasyfikatorem.

* **Model 1: Naiwny Klasyfikator Bayes'a (GaussianNB):** Model probabilistyczny, który zakłada (naiwnie), że cechy są od siebie niezależne. Wariant Gaussian jest odpowiedni dla danych, w których cechy ciągłe mają rozkład zbliżony do normalnego.
* **Model 2: Drzewo Decyzyjne (DecisionTreeClassifier):** Model oparty na regułach, który tworzy strukturę drzewa decyzyjnego do klasyfikacji danych. Jest to model wysoce interpretowalny.

Oba modele zostały wytrenowane na zbiorze treningowym za pomocą metody .fit().

### 3. Wyniki i Ewaluacja

Po zakończeniu treningu, skuteczność obu modeli została oceniona na zbiorze testowym. Wyniki przedstawiono poniżej.

#### Metryki Oceny

* **Dokładność (Accuracy):** Procent poprawnie sklasyfikowanych próbek.
* **Precyzja (Precision):** Stosunek poprawnych predykcji pozytywnych do wszystkich predykcji pozytywnych. Odpowiada na pytanie: "Jaki procent klientów, których model oznaczył jako subskrybentów, faktycznie nimi jest?".
* **Czułość (Recall):** Stosunek poprawnych predykcji pozytywnych do wszystkich rzeczywistych przypadków pozytywnych. Odpowiada na pytanie: "Jaki procent wszystkich faktycznych subskrybentów model był w stanie wykryć?".
* **F1-Score:** Średnia harmoniczna precyzji i czułości, użyteczna metryka dla niezbalansowanych klas.
* **Macierz pomyłek (Confusion Matrix):** Tabela wizualizująca skuteczność modelu.
* **Krzywa ROC:** Graficzna reprezentacja zdolności diagnostycznej klasyfikatora.

#### Analiza Wyników

* **Naiwny Klasyfikator Bayes'a:**
  + **Macierz pomyłek:**
    - Prawdziwie Negatywne (TN): 2
    - Fałszywie Pozytywne (FP): 71
    - Fałszywie Negatywne (FN): 5
    - Prawdziwie Pozytywne (TP): 172
  + **Obserwacje:** Model ten wykazuje bardzo dziwne i prawdopodobnie niepoprawne zachowanie. Klasyfikuje on niemal wszystkie przypadki jako pozytywne ('Yes'). Ma bardzo wysoką czułość (recall) dla klasy 'Yes', ale katastrofalnie niską precyzję. Jest to model bezużyteczny w praktyce.
* **Drzewo Decyzyjne:**
  + **Macierz pomyłek:**
    - Prawdziwie Negatywne (TN): 23
    - Fałszywie Pozytywne (FP): 50
    - Fałszywie Negatywne (FN): 65
    - Prawdziwie Pozytywne (TP): 112
  + **Obserwacje:** Model Drzewa Decyzyjnego, choć daleki od doskonałości, wykazuje znacznie bardziej zbalansowane wyniki. Jest w stanie poprawnie zidentyfikować zarówno przypadki pozytywne, jak i negatywne, choć wciąż popełnia znaczną liczbę błędów obu typów.

### 4. Dyskusja i Wnioski

Porównanie obu modeli prowadzi do jednoznacznych wniosków. **Model Drzewa Decyzyjnego, mimo stosunkowo niskiej ogólnej skuteczności, znacząco przewyższa Naiwny Klasyfikator Bayes'a w kontekście tego konkretnego problemu i zbioru danych.** Model Bayes'a w przedstawionej formie uległ załamaniu, prawdopodobnie z powodu niespełnienia jego fundamentalnych założeń (np. o niezależności cech lub ich rozkładzie) na tym zbiorze danych.

Głównym ograniczeniem badania jest sam zbiór danych. Mała liczba cech oraz ich charakter (szerokie kategorie wiekowe) mogą być niewystarczające do zbudowania precyzyjnego modelu predykcyjnego.

Poniższy wykres krzywej ROC wizualnie potwierdza obserwacje – pole pod krzywą (AUC) jest znacznie większe dla Drzewa Decyzyjnego, co świadczy o jego lepszej zdolności do rozróżniania między klasami.

**Wnioski końcowe:**

1. Zastosowane techniki przygotowania danych (standaryzacja, kodowanie one-hot) są standardem w branży i zostały zaimplementowane poprawnie.
2. Model Drzewa Decyzyjnego dostarczył użytecznych, choć niedoskonałych, wyników, co czyni go lepszym wyborem dla tego zadania.
3. Wyniki modelu Naiwnego Bayes'a wskazują na jego nieadekwatność dla tego konkretnego problemu lub potrzebę zastosowania innego wariantu (np. BernoulliNB lub MultinomialNB, które są przeznaczone dla cech dyskretnych).
4. W celu poprawy skuteczności predykcji, rekomenduje się w przyszłości wzbogacenie zbioru danych o dodatkowe, bardziej informacyjne cechy (np. historia interakcji klienta, szczegółowe dane transakcyjne).