

## **Raport dotyczący kodu do predykcji za pomocą XGBoost**

### **Wstęp**

W tym raporcie omówiono kod przeznaczony do predykcji na zbiorze testowym przy użyciu modelu XGBoost. Kod skupia się na załadowaniu danych, przygotowaniu ich do modelowania oraz budowie modelu.

*Celem* tego raportu jest przedstawienie procesu predykcji na zbiorze testowym przy wykorzystaniu modelu XGBoost. W kolejnych sekcjach omówione zostaną etapy przygotowania danych, budowy modelu oraz eksportu wyników predykcji.

### **1. Załadowanie niezbędnych bibliotek i danych**

W pierwszej części kodu importowane są niezbędne biblioteki, takie jak `xgboost`, `caret` oraz `glmnet`. Następnie dane treningowe i testowe są importowane z plików CSV za pomocą funkcji `fread` lub `read.csv`.

### **2. Przygotowanie danych do modelowania**

Dane treningowe są przekształcane do formatu wymaganego przez XGBoost za pomocą funkcji `xgb.DMatrix`. Parametry modelu, takie jak `max.depth`, `eta`, `nround`, `early_stopping_rounds` oraz `print_every_n`, są zdefiniowane przed trenowaniem modelu.

### **3. Budowa modelu**

Model jest budowany przy użyciu funkcji `xgb.train`, która przyjmuje dane treningowe oraz zdefiniowane wcześniej parametry. Model jest trenowany na danych treningowych, a wynikowy model jest zapisywany do zmiennej `bst_slow`.

### **4. Predykcja na zbiorze testowym**

Dla danych testowych tworzona jest macierz cech za pomocą funkcji `xgb.DMatrix`, a następnie dokonywane są predykcje na podstawie wcześniej wytrenowanego modelu. Wyniki predykcji są zapisywane do obiektu `predicted_df`.

## **5. Eksport wyników predykcji**

Ostatecznie, wyniki predykcji są eksportowane do pliku CSV za pomocą funkcji `write.csv`. Kolumna zawierająca predykcje jest nazwana "Expected", a numerowanie wierszy zaczyna się od 0!

## **Podsumowanie**

Kod do predykcji za pomocą modelu XGBoost został skutecznie zaimplementowany. Proces obejmuje załadowanie danych, przygotowanie ich do modelowania, budowę modelu oraz eksport wyników predykcji. Dzięki zastosowaniu odpowiednich bibliotek i parametrów modelu, uzyskujemy skuteczne i efektywne predykcje na zbiorze testowym.