

Eksploracja danych - etap 2

Krzysztof Nasuta 193328, Filip Dawidowski 193433, Aleks Iwicki 193354

1. Charakterystyka zbioru

- Pochodzenie: [Kaggle](#)
- Liczba przykładów: 5819080
- Format: CSV (3 pliki: `flights.csv` - właściwy zbiór, `airports.csv` - informacje o lotniskach, `airlines.csv` - informacje o liniach lotniczych)
- Ilość zbiorów danych: 1

2. Wprowadzenie

Dataset: **2015 Flight Delays and Cancellations**

Cel: Budowa modelu predykcyjnego klasyfikującego opóźnienia lotów (`ARRIVAL_DELAY` > 15 minut)

Opóźnienia lotów mają znaczący wpływ na funkcjonowanie transportu lotniczego. Niniejszy projekt ma na celu stworzenie modelu uczenia maszynowego przewidującego opóźnienia.

Kluczowe pytania badawcze:

- Które czynniki najsilniej wpływają na opóźnienia?
- Który algorytm osiąga najlepsze wyniki?

3. Założenia wstępne

Podczas przewidywania opóźnień lotów nie będziemy uwzględniać informacji, które nie są dostępne w momencie planowania lotu, takich jak:

- `DEPARTURE_TIME` (nie mylić z `SCHEDULED_DEPARTURE`)
- `DEPARTURE_DELAY`
- `TAXI_OUT`
- `WHEELS_OFF`
- `ELAPSED_TIME`
- `AIR_TIME`
- `WHEELS_ON`
- `TAXI_IN`
- `ARRIVAL_TIME`
- `ARRIVAL_DELAY`

Spowoduje to znaczne obniżenie dokładności modeli, lecz pozwoli na realistyczne przewidywanie.

4. Przygotowanie Danych

Źródła danych:

- `flights.csv`
- `airlines.csv`
- `airports.csv`

Kroki przetwarzania:

1. Ładowanie danych z pliku `flights.csv` z ograniczeniem do 250,000 rekordów

2. Definicja zmiennej celu: **DELAYED = 1** jeśli **ARRIVAL_DELAY > 15**
4. Balansowanie zbioru danych - równa liczba opóźnionych i nieopóźnionych lotów
6. Podział zbalansowanych danych na zbiór treningowy (80%) i testowy (20%)

Cechy wykorzystane w modelu:

- Kategoryczne: AIRLINE, ORIGIN_AIRPORT, DESTINATION_AIRPORT, DAY_OF_WEEK, MONTH
- Numeryczne: YEAR, DAY, FLIGHT_NUMBER, SCHEDULED_DEPARTURE, SCHEDULED_TIME, DISTANCE, SCHEDULED_ARRIVAL

Usunięte cechy (data leakage): DEPARTURE_TIME, DEPARTURE_DELAY, TAXI_OUT, WHEELS_OFF, ELAPSED_TIME, AIR_TIME, WHEELS_ON, TAXI_IN, ARRIVAL_TIME, ARRIVAL_DELAY

5. Metodologia

Wykorzystane modele:

| Model | Implementacja |
|------------------------|---|
| „Drzewo Decyzyjne” | DecisionTreeClassifier(random_state=42) |
| „Las Losowy” | RandomForestClassifier(n_estimators=100, random_state=42) |
| „Regresja Logistyczna” | LogisticRegression(random_state=42, max_iter=1000) |
| „K-NN” | KNeighborsClassifier(n_neighbors=5) |
| „Sieć Neuronowa” | MLPClassifier(hidden_layer_sizes=(100,50), max_iter=500) |

Optymalizacja hiperparametrów (Algorytm Genetyczny):

- Populacja: 15 osobników
- Generacje: 15
- Krzyżowanie: dwupunktowe (prawdopodobieństwo 0.5)
- Mutacja: gaussowska (prawdopodobieństwo 0.2, $\sigma=0.1$)
- Selekcja: turniejowa (rozmiar turnieju = 3)
- Fitness: średnia dokładność z 3-krotnej walidacji krzyżowej

Parametry optymalizowane:

| Model | Parametr | Zakres |
|-----------------------|---------------------|-------------|
| „Random Forest” | n_estimators | 10-200 |
| | max_depth | 3-20 |
| | min_samples_split | 2-20 |
| | min_samples_leaf | 1-10 |
| „Logistic Regression” | C | 0.01-100.0 |
| | max_iter | 100-2000 |
| „Neural Network” | hidden_layer_size_1 | 50-200 |
| | hidden_layer_size_2 | 20-100 |
| | alpha | 0.0001-0.01 |
| | max_iter | 200-1000 |

6. Wyniki Eksperymentów

6.1. Wyniki modeli z domyślnymi parametrami

Dataset: x rekordów (zbalansowany)

| Model | Dokładność | Cechy |
|-----------------------|------------|-------|
| „Decision Tree” | `x` | `x` |
| „Random Forest” | `x` | `x` |
| „Logistic Regression” | `x` | `x` |
| „K-NN” | `x` | `x` |
| „Neural Network” | `x` | `x` |

6.2. Wyniki optymalizacji genetycznej

Najlepsze parametry znalezione przez algorytm genetyczny:

| Model | Najlepsze parametry | CV Score | Rozmiar danych |
|-----------------------|---------------------|----------|----------------|
| „Random Forest” | `x` | `x` | `x` |
| „Logistic Regression” | `x` | `x` | `x` |
| „Neural Network” | `x` | `x` | `x` |

6.3. Analiza ważności cech

Ranking ważności cech (Random Forest):

| Pozycja | Cecha |
|--------------|-------|
| Ważność „1.” | |
| `x` | `x` |
| „2.” | `x` |
| `x` | „3.” |
| `x` | `x` |
| „4.” | `x` |
| `x` | „5.” |
| `x` | `x` |

6.4. Analiza wpływu usuwania cech

Wpływ usunięcia poszczególnych cech na dokładność:

| Scenariusz | Random Forest | Logistic Regression | Neural Network | Liczba cech |
|---------------------------|---------------|---------------------|----------------|-------------|
| „Wszystkie cechy” | `x` | `x` | `x` | `x` |
| „Bez AIRLINE” | `x` | `x` | `x` | `x` |
| „Bez ORIGIN_AIRPORT” | `x` | `x` | `x` | `x` |
| „Bez DESTINATION_AIRPORT” | `x` | `x` | `x` | `x` |
| „Bez DISTANCE” | `x` | `x` | `x` | `x` |

| | | | | |
|-----------------------------|-----|-----|-----|-----|
| „Bez informacji o lotnisku” | `x` | `x` | `x` | `x` |
| „Bez informacji czasowych” | `x` | `x` | `x` | `x` |
| „Tylko kategoryczne” | `x` | `x` | `x` | `x` |
| „Tylko numeryczne” | `x` | `x` | `x` | `x` |

Najlepsze scenariusze dla każdego modelu:

- Random Forest: x (dokładność: x)
- Logistic Regression: x (dokładność: x)
- Neural Network: x (dokładność: x)

7. Wnioski i obserwacje

Wpływ balansowania danych:

- Przed balansowaniem: x opóźnionych, x nieopóźnionych lotów
- Po balansowaniu: x opóźnionych, x nieopóźnionych lotów
- Wpływ na dokładność: x

Najważniejsze cechy wpływające na opóźnienia:

1. x
2. x
3. x

Optymalizacja genetyczna vs. domyślne parametry:

- Średnia poprawa dokładności: x%
- Najlepsza poprawa: x dla modelu x

8. Rekomendacje

1. **Model produkcyjny:** x z parametrami x
2. **Kluczowe cechy:** Skupić się na x, x, x
3. **Dalsze badania:**
 - Testowanie na pełnym zbiorze danych
 - Dodanie cech pogodowych
 - Analiza sezonowości opóźnień

9. Podsumowanie

Projekt wykazał skuteczność x w predykcji opóźnień lotów. Algorytm genetyczny pozwolił na x poprawę wyników względem parametrów domyślnych. Najważniejszymi czynnikami wpływającymi na opóźnienia okazały się x.

Osiągnięte cele:

- Dokładność najlepszego modelu: x%
- Identyfikacja kluczowych cech
- Optymalizacja hiperparametrów
- Analiza wpływu poszczególnych cech

Kod źródłowy:

- model_comparison.py - porównanie modeli z domyślnymi parametrami
- genetic_tuning.py - optymalizacja genetyczna

- `benchmark_analysis.py` - analiza wpływu cech